

1184-3170  
NASA Contractor Report 175270

A Method to Combine Remotely  
Sensed and Insitu Measurements:  
Program Documentation

E. L. Peck, E. R. Johnson and M. Y. Wong

May 1984—Interim Report



National Aeronautics and  
Space Administration  
Goddard Space Flight Center  
Greenbelt Road  
Greenbelt, Maryland 20771

# **NASA Contractor Report 175270**

## **A Method to Combine Remotely Sensed and Institu Measurements: Program Documentation**

**E. L. Peck, E. R. Johnson and M. Y. Wong**  
*Hydex Corporation*  
**Suite 502**  
**11150 Main Street**  
**Fairfax, Virginia 22030**

**Prepared for:**  
**Goddard Space Flight Center**  
**Greenbelt Road**  
**Greenbelt, Maryland 20771**

**NASA**  
National Aeronautics  
and Space Administration

**Scientific and Technical  
Information Office**

**1984**

## PREFACE

This is the first report of the third phase of a study by the Hydex Corporation for the National Aeronautics and Space Administration (NASA). The study is funded under the Conservation and Pollution section of the AgRISTARS program, a joint program for Agricultural and Resources Inventory Surveys Through Aerospace Remote Sensing.

The study is to develop and test technology to improve hydrologic modeling by incorporating remotely sensed measurements in operational procedures.

The first phase of the study determined the suitability of present and planned remote sensing capabilities for commonly used hydrologic models.

The second phase included the development of methods to use remotely sensed measurements with hydrologic models to ensure that simulated results are consistent with observed conditions. As part of this phase a (correlation area) method was developed to combine remotely sensed and standard measurements to obtain improved areal averaged estimates of hydrologic variables.

The objective of this third phase is to provide user and programmer information to implement the techniques developed during Phase II. This report presents such information for the correlation area method (CAM).

## TABLE OF CONTENTS

	Page
PREFACE.....	ii
TABLE OF CONTENTS.....	iii
LIST OF ILLUSTRATIONS.....	iv
CHAPTER I. CORRELATION AREA METHOD.....	1
INTRODUCTION.....	1
CRITERIA FOR METHOD.....	5
Illustration of CAM Method.....	6
DATA REQUIREMENTS.....	10
APPLICATION OF METHOD.....	13
DOCUMENTATION.....	14
CHAPTER II. PROGRAM MODIFICATION.....	15
INTRODUCTION.....	15
Basin Boundary Convention.....	15
The Grid Method.....	15
Areal Sample Processing.....	16
Areal Samples.....	18
Imagery Samples.....	19
CHAPTER III. PROGRAM INPUT AND OUTPUT.....	22
INTRODUCTION... ..	22
General Format.....	22
Specific Format (Field Definitions).....	25
Program Output.....	28
Sample Run.....	31
CHAPTER IV. AIDS TO USERS.....	35
PROGRAM DESCRIPTION.....	35
Boundary Data Array.....	35
Augmented Boundary Flags.....	38
Boundary Data Chains.....	39
Input Data Processing.....	40
ACKNOWLEDGEMENTS.....	43
REFERENCES.....	44
APPENDIX A CAM PROGRAM LISTING.....	A-1
APPENDIX B. ERRATA FOR COMBINING REMOTELY SENSED.....	B-1
AND OTHER MEASUREMENTS FOR HYDROLOGIC AREAL AVERAGES NASA-CR170457 October 31, 1982	

LIST OF ILLUSTRATION

FIGURE 1-1 BRIDGING THE GAP OF UNCERTAINTY AND CORRESPONDENCE

FIGURE 1-2 EXAMPLE OF CORRELATION AREA METHOD FOR ASSIGNING CONVENTIONAL (POINT), AIRCRAFT (LINE) AND SPACE (AREAL) MEASUREMENTS TO REPRESENTATIVE BASIN AREAS

FIGURE 1-3 AREAS ASSIGNED TO POINT, LINE AND AREAL MEASUREMENTS (EXAMPLE)

FIGURE 3-1 A SAMPLE DATA SET

## CHAPTER I

### CORRELATION AREA METHOD

#### INTRODUCTION

The correlation area method (CAM) was developed to test technology for improving hydrologic modeling by incorporating remotely sensed measurements in operational procedures. Hydex published four National Aeronautics and Space Administration (NASA) contractor reports on the results of the study:

1. Review of Hydrologic Models for Evaluating Use of Remote Sensing Capabilities<sup>(1)</sup>;
2. Strategies for Using Remotely Sensed Data in Hydrologic Models<sup>(2)</sup>;
3. Combining Remotely Sensed and Other Measurements for Hydrologic Areal Averages<sup>(3)</sup>; and
4. Creating a Bridge Between Remote Sensing and Hydrologic Models<sup>(4)</sup>.

The initial report surveyed seven commonly used hydrologic models and presented information on the structure, parameters, states, and required inputs for each model. This report indicates the overall complexity of each model, showing how runoff, soil moisture and other processes (infiltration, percolation, evaporation, interception, and losses) are modeled. This detailed knowledge of models is required to evaluate how additional information (in this case, remotely sensed measurement of hydrologic variables) might improve the models.

Strategies to us existing and planned remotely sensed information in commonly used hydrologic models are in the second contractors report. This report analyses the measurement characteristics (resolution, error, and precision in space and time) of six variables considered most promising for hydrologic

models. This information, coupled with the structure of the seven models presented in the first report, provide a sound basis for evaluating the usefulness of the six variables for operational modeling. The six variables are:

- (a) soil moisture;
- (b) impervious area;
- (c) land cover;
- (d) areal extent of snow cover;
- (e) areal extent of frozen ground; and
- (f) water equivalent of snow cover.

Hydrologic variables, such as soil moisture and the amount of water in a snow cover (water equivalent), are modeled over time by the hydrologic model. The second report suggested that actual measurements of hydrologic variables, such as soil moisture and water equivalent of the snow cover, might help keep the modeled states of hydrologic variables consistent with actual values in the real world. Remote measurements can provide excellent information on the areal variation of a hydrologic variable; whereas point insitu measurements on the ground generally provide a more accurate measurement at one specific point. Most hydrologic models model the areal average of hydrologic variables. Intuitively the most accurate estimate of the areal average can be obtained using all available information.

The third report presents the fundamental concepts for the Correlation Area Method (CAM) for estimating areal averages of hydrologic variables from a data base with a mixture of measurement technologies. Since this report may not be available, the concepts are reviewed in the following sections of this chapter.

The results of the first phase of the study emphasized several reasons why remotely sensed information has not been used for operational hydrologic modeling. One of these is the dissimilarity (correspondence) in the time and space averages as envisioned by the hydrologic model, as exist in the real world, and as measured by remote sensing systems. A second reason derives from the uncertainties in the estimates obtained from the actual observations and the value envisioned by the model. Thus, there are two sources of information, the observations and the model, both of which have some degree of uncertainty.

The fourth report presented technology to bridge the gap of uncertainty and correspondence between remote sensing and hydrologic models in order to keep our models in tune with the real world. This is called updating a model. Figure 1-1 demonstrates how this bridging is accomplished. The correlation area method is used to produce the best estimate of a hydrologic state (for example, the amount of soil moisture in the surface layer of the soil). Modifications are required in the hydrologic model to achieve correspondence between this real world measurement and the modeled states. Programs with documentation for updating the models of the National Weather Service River Forecast System (NWSRFS) will be published in a forthcoming NASA contract report.

The remainder of this chapter briefly describes the CAM method. For a more complete description the reader is referred to the third NASA report listed above<sup>(3)</sup>.

# BUILDING A BRIDGE

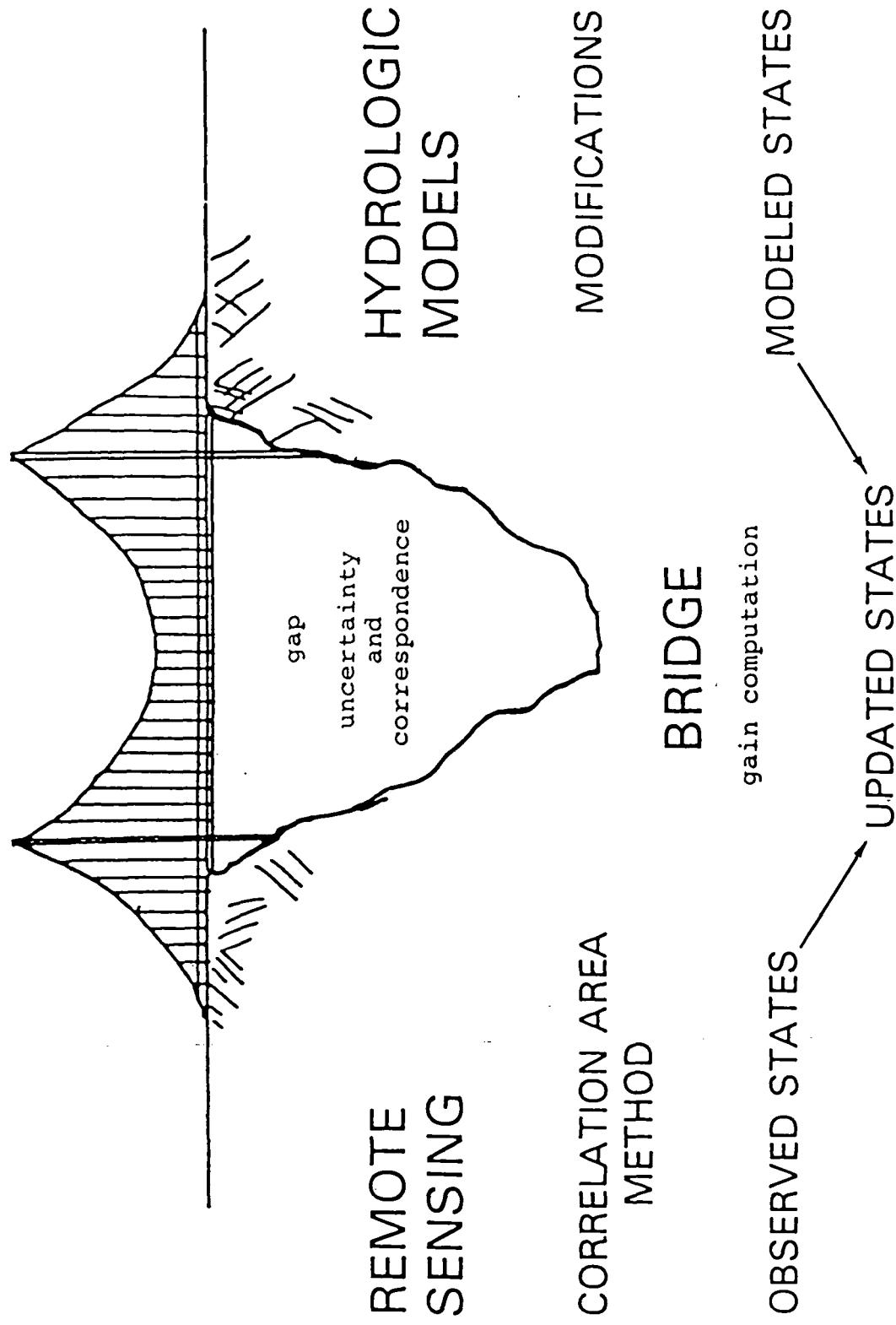


FIGURE 1-1. BRIDGING THE GAP OF UNCERTAINTY AND CORRESPONDENCE

## CRITERIA FOR METHOD

For a technique to be of maximum value for estimating areal averages of hydrological variables from all available measurements, the following criteria should be satisfied:

- a. It should not be dependent on particular measurement technologies;
- b. It should be an objective technique;
- c. It should work regardless of the mix of data available in any one time step;
- d. It should explicitly recognize the sampling geometry of the data;
- e. It should explicitly recognize differences in measurement accuracy of different technologies; and
- f. It must produce some estimate of the accuracy of the areal estimate.

The correlation area method meets the criteria listed above for estimating areal averages from data of various sampling characteristics. The algorithm performs in a desirable fashion giving greater weight to measurements that are more accurate and that cover a larger portion of the basin.

The algorithm is a heuristic procedure (an engineering approach) that takes liberties with a more theoretically correct approach for the simplicity and operational capability.

The weight assigned to each sample in computing an areal average value of the hydrological variable is determined by:

- (a) the normal correlation between that type of measurement and the hydrological variable; and
- (b) the area of the basin that is best represented by that measurement.

### Illustration of CAM Method

To illustrate the basic principles, a simple case follows. There are three measurements for the hydrological variable for the basin in Fig. 1-2: a point, a transect (line) and an areal sample. The areal sample covers more than the basin of interest.

For the purpose of this illustration, the following assumptions are made regarding the correlation of each measurement with the random field of a hydrological variable.

- (a) The areal measurement has a correlation of 0.7 with the true areal average over the sample domain;
- (b) The line measurement has a correlation of 0.85 with a random point along the flight line; that correlation decreases on either side of the line as shown by the values of the parallel dashed lines in Fig. 1-2; and
- (c) The point measurement has a correlation of 0.94 with the variable at the point, decreasing around the point as shown by the values of the circles in Fig. 1-2.

The area of the basin not encompassed within the 0.7 correlation lines for the line measurement and the 0.7 circle for the point measurement indicates that the areal sample provides the best estimate for this area. The areal sample measurements have a 0.7 correlation with the random field of the variable for this area. For the rest of the basin, the point measurement (within the 0.7 correlation circle) and the line measurement (between the 0.7 dashed lines) provide a better estimate than does the areal sample measurement. The areas assigned to each of the three measurements are shown by the shaded areas in Figure 1-3.

The informational value of each measurement is dependent upon how much area is assigned to that measurement and on how well that information correlates with the random variable in that area. Since the correlation of the point and line measurements varies for different portions of these areas, the informational values are not constant over these areas.

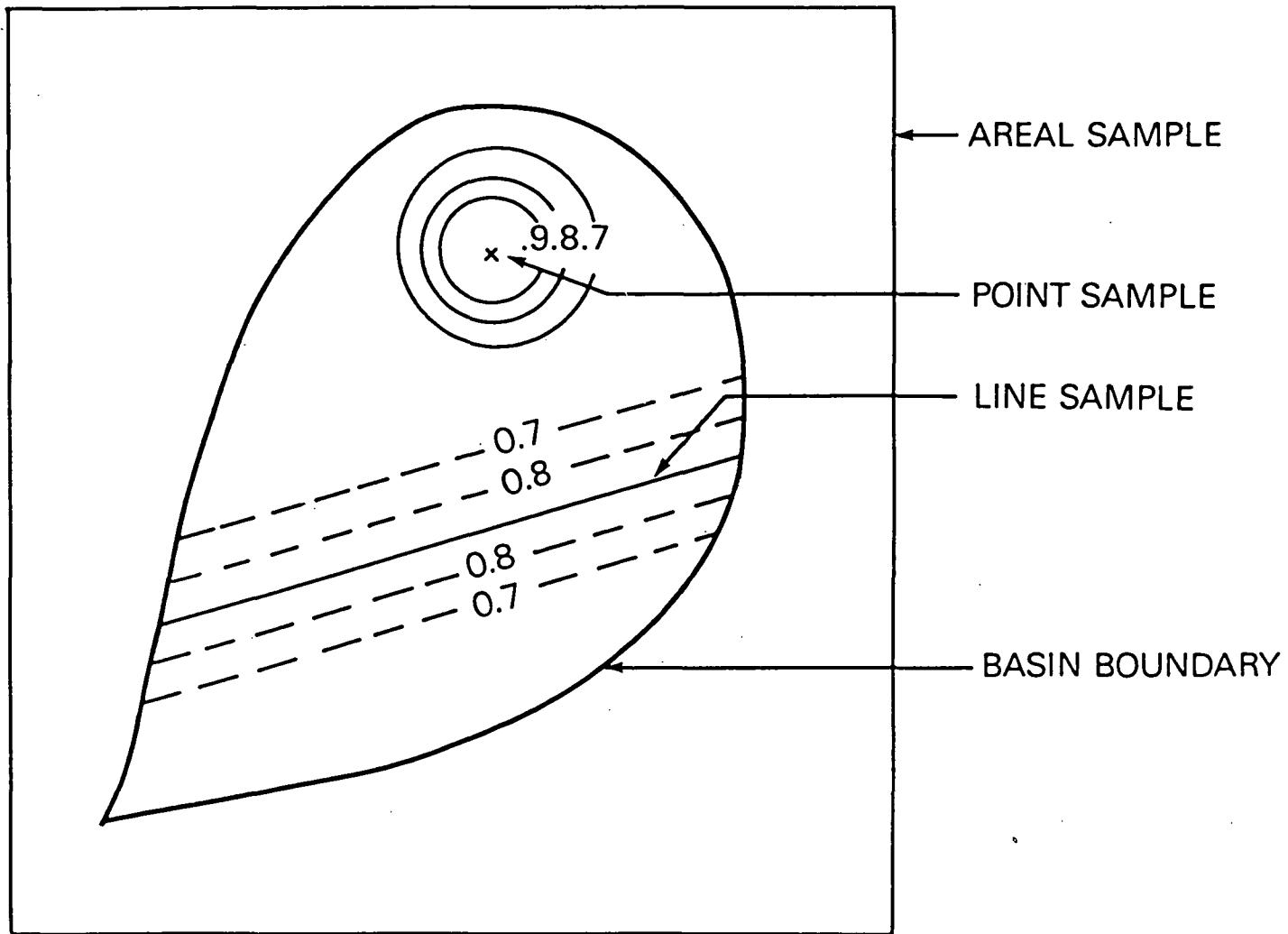


FIG. 1-2. EXAMPLE OF CORRELATION AREA METHOD FOR ASSIGNING CONVENTIONAL (point), AIRCRAFT (line) AND SPACE (areal) MEASUREMENTS TO REPRESENTATIVE BASIN AREAS.

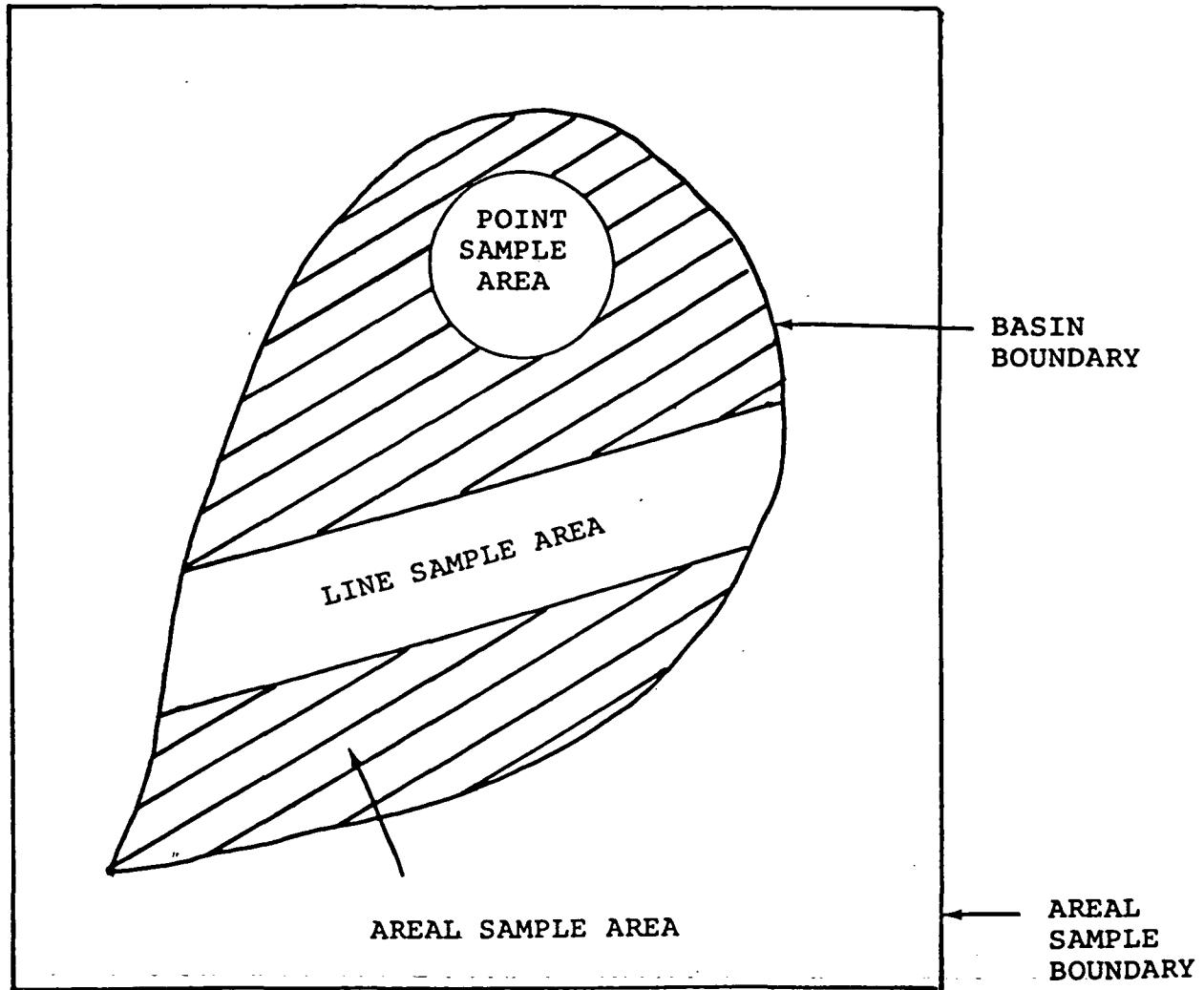


FIG. 1-3. AREAS ASSIGNED TO POINT, LINE AND AREAL MEASUREMENTS (example).

The approach to determining the best average values for the basin includes a weighting for each area. The correlation area is equal to the sums of the products of the correlation multiplied by the portion of the area assigned to that measurement covered by that correlation.

For example, the correlation area for the point sample,  $A_p$ , in the case illustrated in Figures 1-2 and 1-3 is approximately:

$$A_p = 0.92 \times (\text{area between the sample and 0.9 circles}) \\ + 0.85 \times (\text{area between the 0.9 and 0.8 circles}) \\ + 0.75 \times (\text{area between the 0.8 and 0.7 circles})$$

The correlation area for the line measurement,  $A_l$ , is determined by the sum of the products of the average correlation multiplied by the area between each correlation line on either side of the flight line as shown in Fig. 1-2.

In this example, the correlation area for the areal sample,  $A_a$ , is 0.7 times the area of the basin assigned to the areal measurement as shown on Fig. 1-3.

The final estimated areal average value,  $\hat{P}$ , of the variable for the basin is simply a weighted average of the measurements. The weight ( $\lambda_i$ ) for a measurement is equal to its correlation area ( $A_i$ ) divided by the sum of all the correlation areas for the basin.

$$\lambda_i = \frac{A_i}{\sum A_i} \quad (1-1)$$

A measure,  $\omega_m$ , of the overall accuracy of the final estimate ( $\hat{P}$ ) is obtained by dividing the sum of the correlation areas ( $A_i$ ) by actual area of the basin (B).

$$\omega_m = \frac{\sum A_i}{B} \quad (1-2)$$

By its definition  $\omega_m$  must be between 0 and 1 and can be loosely referred to as the correlation of the estimated value ( $\hat{P}$ ) with the actual average ( $P$ ) of the parameter over the basin.

The algorithm has an intuitively reasonable behavior: more accurate measurements get a larger sample area, a larger correlation area, and, thus, a larger weight than less accurate samples. The location of the samples also affect the shape of the sample area.

When multiple line, point and areal samples are considered, the shape of the sample areas becomes considerably more complex than for the simple case shown in Fig. 1-2, but the basic concept is unchanged. Details on the algorithms to handle complex cases are described by Johnson et al<sup>(3)</sup>.

#### DATA REQUIREMENTS

Applying the correlation area method requires these statistical characteristics of each measurement type in order to properly weight it:

- (a) the correlation of a point measurement with the true value ( $C_p$ );
- (b) the correlation of a transect (line) measurement with the true value at a point along the line ( $C_l$ );
- (c) the correlation of an areal average with the true areal average ( $C_a$ );
- (d) the rate at which the point or line correlation decays with distance ( $\alpha$ ); and
- (e) the measure of randomness in an areal distributed random field ( $\sigma$ ).

The sampling correlation functions listed above require that several covariance parameters be estimated to implement the correlation area technique. These covariance parameters relate both to the accuracy of particular measurement technologies and to the correlation decay in space of the random field under study.

A brief discussion pertaining to each of the data requirements listed above follows:

Point, Sample Correlation,  $C_p$

The value of  $C_p$  is the correlation of a point measurement with the true value at that point. Based on investigations of the accuracy of ground sampling technologies, it should be relatively easy to estimate  $C_p$ . Furthermore, it is expected that  $C_p$  will be generally "large", say, approximately 0.9 or better, for most technologies.

Line Sample Correlation,  $C_\ell$

The value of  $C_\ell$  is the correlation of a flight-line sample with the value at a point randomly located along the flight line. Presumably, historical ground truth experiments will be sufficient to estimate  $C_\ell$ .  $C_\ell$  actually mixes two effects: first, the accuracy of the technology as it measures the true flight-line average; and the second the correlation of a point along the flight line with the flight-line average. A more accurate technology will increase  $C_\ell$ , but even a perfect measurement technology would not guarantee a  $C_\ell$  of 1.0 because of the variability of point values along the flight line. Thus,  $C_\ell$  should be expected to be less than  $C_p$ .

Areal Sample Correlation,  $C_a$

The value of  $C_a$  is the correlation of an areal sample with the true areal average over the sampling domain. Presumably, ground truth experiments will provide information to estimate  $C_a$ , but some degree of subjectivity may be needed as well.

Decay Parameter,  $\alpha$

The variable under study is assumed to be a random field. This random field, whether soil moisture, snow water equivalent,

or whatever, is assumed to have a homogeneous mean value and variance and isotropic covariance in space described by a simple exponential covariance function. The decay parameter  $\alpha$  describes the value of information at one location in estimating the random field at another site.

There are three possible approaches for estimating the value of  $\alpha$ : the use of historical data, real-time data, and a conceptual model. However, the historical approach and the conceptual model approach offer the most promise for operational estimates of  $\alpha$ .

Historical Data. Historical data on soil moisture or snow water equivalent can be analyzed to estimate  $\alpha$ . Then the value of  $\alpha$  can be assumed to apply to current conditions. The difficulties with this approach are: (a) procuring a historical data base, and (b) developing a procedure to "stratify" the data for different values of  $\alpha$  related to some easy-to-identify property of current conditions.

Real Time Data. If enough point data are available in real time, a value of  $\alpha$  can be estimated for the current condition of the random field. Using remotely sensed data for this approach is rather difficult because of the sample averaging properties of this type of data.

Conceptual Model. The conceptual model approach can be illustrated by considering soil moisture as the product of two random fields: the field capacity and the fraction of field capacity that is filled. The variability of the field capacity can be related to a soil map. If a conceptual model is developed to relate the statistics of the "fraction of field capacity" to some easy to estimate parameters (e.g., the antecedent precipitation index), it may be possible to estimate  $\alpha$  for the soil moisture without actual measurements of soil moisture.

### Measure of Randomness, $\sigma$

The same three approaches for estimating the decay parameter,  $\alpha$ , can be employed to estimate the randomness measure,  $\sigma$  which is the standard deviation of the random field under study.

### Data Base Availability

The data base to compute the statistical parameters for the correlation area method is often unavailable. However, for the agricultural areas of the north central plains area of the United States and for the central area of Russia there are sufficient historical data to derive initial estimates to use the correlation area method for soil moisture and water equivalent of the snow cover. Of course, the data base from remotely sensed measurements of hydrological variables increases there will be more information to estimate the required statistical parameters.

### APPLICATION OF METHOD

The primary application for which the correlation area method was developed was to provide a means to integrate measurements from conventional hydrological networks and remote sensing for use in operational hydrology. However, the resulting improved estimates of the areal averages of hydrological variables potentially have many other uses. For example, such improved estimates of areal averages of soil moisture may be used with large scale climatic models or agricultural models for predicting droughts and crop yields.

The method can be used in a reverse sense for network design to determine what combined data networks would be required to enhance a model's performance. For example, the accuracy of areal estimates of the water equivalent of the snow cover based on point (insitu) measurements are often poor. Having a knowledge of the accuracy of remote sensing techniques (i.e., aerial gamma radiation surveys, Carroll et al.<sup>(5)</sup> or microwave, Schmugge<sup>(6)</sup>) and an analysis of the data accuracy required for a specific improvement in model performance, the correlation area method

could be applied (in reverse) to define the mix of measurement technologies networks required to produce the desired improvement.

#### DOCUMENTATION

The remainder of this report consists of a program to implement the correlation area method and user and programmer information.

Chapter 2 describes changes and enhancements to the CAM as it was presented in the original NASA Contractor Report<sup>(3)</sup>. These modifications were made primarily in the interest of computational efficiency and to simplify the user input. Chapter 3 describes the input and output for the program, including a sample run. For those interested in using the program information on processing input data and on the major subroutines see Chapter 4. Appendix A contains a complete listing of the FORTRAN program for implementing CAM. An errata sheet for the contractor report<sup>(3)</sup> on CAM is given in Appendix B.

## CHAPTER II

### PROGRAM MODIFICATION

#### INTRODUCTION

In the interest of program efficiency, ease of use and compatibility with other programs, a number of modifications to the original CAM algorithm have been made to implement it as a computer code. This chapter describes these modifications.

#### Basin Boundary Convention

The original CAM report adopted a counterclockwise convention for the description of boundaries as pairs of points. Since the National Weather Service River Forecast System (NWSRFS) has adopted a clockwise convention for their basin boundaries, the CAM program will do likewise. This trivial change affects only the sign of computed areas (see Appendix I of original report) and makes the "forward" direction in Appendix H of the original report clockwise rather than counterclockwise.

#### The Grid Method

The original CAM report describes procedures to determine the boundary of the area within the basin associated with each sample which is most highly correlated with that sample--the "sample area". Then the appropriate sampling correlation function is integrated over the region within the sample area to produce a weighted sample area--the "correlation area"--for each sample. For most sample area boundaries the required integration cannot be performed analytically, thus the computed correlation area is only approximate--the degree of approximation depends on the accuracy of the numerical integration step.

An alternate method, the grid method, overlays the basin with a uniform square grid. Each grid point is allocated to the measurement with which it is most highly correlated and assigned a grid weight equal to the magnitude of that correlation. To find the basin wide areal average the sample values are weighted by the sum of the grid weights allocated to the sample and divided by the total sum of all grid weights. The accuracy measure is simply the sum of all grid weights divided by the total number of grid points on the basin. As the mesh spacing in the grid is decreased, the grid method will converge to the same results as the original CAM report described.

The grid method has several advantages. It avoids, for the most part, the tedious computations needed to define each sample area boundary. It is comparatively easier to implement as a computer code. Its accuracy is directly related to the same parameter which will control its computational demands--the mesh size. Its numerical accuracy depends on a single parameter, the mesh size, rather than choices for the number of terms used to approximate the multitude of integrals resulting from the original method. (It is likely as well that one would choose for expediency to approximate the circular and parabolic boundaries of the original report with linear approximations which would also introduce additional numerical accuracy parameters.)

In short, the grid method appears to be a better computational approach with which to implement CAM though it is less analytically elegant than the original sample area approach.

#### Areal Sample Processing

As envisioned in the original report, all areal-average samples would be processed in the same manner regardless of the spatial resolution of the remote sensing technology employed. The basin has to be divided into sub-basins each within the domain of a single areal sample. If two technologies were employed the basin would be further subdivided to account for all the overlapping domains of samples. These techniques work fine

for samples with large spatial size as envisioned in the original report--especially if only one or two technologies were employed at any time. But if high-resolution data and multiple technologies are processed in this way thousands, perhaps millions, of sub-basins will be created.

To avoid the computational burden of millions of sub-basins, a two-step procedure will be used for processing areal sampling technologies. The first step is to combine all the data from each individual technology to estimate the areal average for the entire basin. The second step is to combine the estimates from all areal sampling technologies into a single estimate for the entire basin. The final result of this two-step process is almost identical to the original procedure and it is much more efficient.

A further reduction in computational demands can be achieved by making a distinction between "areal samples" and "imagery". When the spatial resolution of the technology is small enough it becomes absurd to consider a pixel to be partially within the basin -- at this point the technology is producing an imagery type areal samples. The "imagery" type data is input by specifying the center point of each pixel and a value for each pixel rather than locating the four corners of each sample value as is required for the "areal sample" type data. This considerably reduces input requirements for high-resolution data types. For the imagery type, pixels are considered to be wholly within or outside the basin if their center point alone is within or outside the basin. The two-step procedure is described in some detail below.

STEP ONE: Combine All Samples From One Technology.

The objective of this step is to combine the areal samples from a single technology i to find an estimate  $M_{A_i}$  of the areal

average over the entire basin and also estimate the correlation  $C_{A_i}$  of  $M_{A_i}$  with the actual basin-wide average. There are two alternate procedures used--one for areal samples and one for imagery samples as described above.

#### Areal Samples

First the correlation of each areal sample value must be adjusted for partial coverage of the basin via equation (2-1) below

$$\hat{C}_{ij} = C_{ij} (B \cap D_{ij}) / D_{ij} \quad (2-1)$$

where  $C_{ij}$  = correlation of  $j$ -th areal sample of the  $i$ -th technology with the true areal average for the  $j$ -th area (probably the same for all  $j$ ).

$D_{ij}$  = the area ( $Km^2$ ) of the  $j$ -th areal sample of the  $i$ -th technology.

$(B \cap D_{ij})$  = the areal ( $Km^2$ ) within the basin and within  $D_{ij}$

$\hat{C}_{ij}$  = adjusted correlation of the  $j$ -th sample.

Then the areal average  $M_{A_i}$  over the basin can be estimated from the individual sample values equation

$$M_{A_i} = \frac{\sum_j \hat{C}_{ij} M_{ij}}{\sum_j \hat{C}_{ij}} \quad (2-2)$$

Finally, the correlation of the true areal value over the basin with the estimate  $M_{A_i}$  can be estimated via:

$$C_{A_i} = \frac{\sum_j \hat{C}_{ij} D_{ij}}{B} \quad (2-3)$$

where  $B$  = basin area.

### Imagery Samples

In fact, imagery samples can be processed in the same way that areal samples are processed as described above. However, a significant simplification results from recognizing that (for imagery only):

$$(B \cap D_{ij}) = D_{ij} \text{ for } j \text{ inside basin}$$

$$= 0 \text{ for } j \text{ outside basin}$$

because the pixels of imagery data are not considered to be partially in the basin. This reduces equation (2-2) above to the form:

$$M_{A_i} = \frac{\sum_{j \text{ in basin}} C_{ij} M_{ij}}{\sum_{j \text{ in basin}} C_{ij}} \quad (2-4)$$

Further, if  $C_{ij}$  is the same for all  $j$  pixels (which is the expected case) then equation (2-4) is simply

$$M_{Ai} = \frac{1}{J} \sum_{\substack{j \text{ in} \\ \text{basin}}} M_{ij} \quad (2-5)$$

where  $J$  = the number of pixels in the basin.

Similarly equation (2-3) reduces to

$$C_{Ai} = C_i \quad (2-6)$$

where  $C_i = C_{ij}$  is the common correlation of each pixel with the true areal average over the domain of the pixel.

#### STEP TWO: Combine All Technologies

The second step in processing areal sampling technologies is to combine the  $N$  estimates  $M_{Ai}$  ( $i=1, \dots, N$ ) of the basin-wide average into a single estimate  $M_A$  and also to find  $C_A$ , the associated correlation measure. By expanding the procedures described in the original CAM report (Appendix F) to the  $N$  measurement technologies the following are obtained:

$$M_A = \sum_{i=1}^N b_i M_{Ai} \quad (2-7)$$

$$b_i = \frac{1}{\left(\frac{1-c_{A_i}}{c_{A_i}}\right) \sum_{j=1}^N \left(\frac{c_{A_j}}{1-c_{A_j}}\right)} \quad (2-8)$$

$$c_A = \frac{1}{1 + \sum_{i=1}^N \left(\frac{b_i^2 (1-c_{A_i})}{c_{A_i}}\right)} \quad (2-9)$$

which completes the areal sample processing steps.

## CHAPTER III

### PROGRAM INPUT AND OUTPUT

#### INTRODUCTION

Before running the program a data file containing the basin data and sample values must be set up. The name of this data file is supplied to the program in response to a prompt.

The input data file consists of a sequence of card images in a fixed field format. First, the general format for these card images will be described, then the field definitions for each data item will be given.

#### General Format

The data file card images are of two types: control cards and data cards. The control cards contain a character in column one, called a control code, and are otherwise blank. The control code tells the program what to expect on the next few data cards. The following is a list of control codes, with a brief description of their meaning:

<u>Control Code</u>	<u>Meaning</u>
*	The following data cards, are comments. They will be echo-printed, but are otherwise ignored. Another * control code ends the comments.
C	The following data cards contain comments, but these comments will be printed on the output as a title.
B	The following data cards define the <u>basin</u> .
N	Begin a <u>new</u> set of data. The following data cards identify the data set.
E	<u>End</u> of a data set.

<u>Control Code</u>	<u>Meaning</u>
S	<u>Stop.</u> End of all data sets.
P	The following data cards contain <u>point</u> data.
L	The following data cards contain <u>line</u> data.
A	The following data contain <u>areal</u> data.
I	The following data cards contain <u>image</u> data.

A typical data set will look as follows:

*	comment lines
*	
C	title to be printed on the output
B	basin boundary data
N	
P	point data
L	flight-line data
A	areal data
A	areal data (2nd technology)
E	
C	
	new title to identify second set of data
N	
P	point data
I	imagery data
A	areal data
E	
S	indicates end of run

Data cards must have a blank in column 1. The data cards between two control cards are to be thought of as an input group. The format for the group is determined by the control code that precedes it. The general format for each group follows.

Basin Boundary Group: A sequence of latitude-longitude pairs that defines the basin boundary in a clockwise

manner. This input group must precede other data for this case. If no basin data is given, the previous basin data is used.

Point Data Group: This group of input contains all of the point data for a data case. Each point value is described by:

- a 12-character identifier
- latitude
- longitude
- sample value
- correlation at zero distance
- decay parameter in  $\text{Km}^{-1}$

If the correlation value or decay parameter are the same as on the previous data card, then they need not be repeated.

Line Data Group: This input group is similar in format to point data. For each line sample the following data is supplied.

- a 12-character identifier
- latitude and longitude of start of line
- latitude and longitude of end of line
- sample value
- zero distance correlation
- decay parameter in  $\text{Km}^{-1}$

Here too, if correlation or decay values are the same for several lines, these values need not be repeated on the card images.

Areal Sample Group: Each areal sample group gives input for a single areal sampling technology. Sample values are assumed to represent nonoverlapping quadrilateral areas. The technology as a whole should have a 16-character descriptive title and a (default) value for the correlation of each areal sample with the true areal value.

Each individual sample then requires:

- the sample value (numeric)
- the correlation measure (optional)
- (latitude, longitude) of 1st corner
- (latitude, longitude) of 2nd corner
- (latitude, longitude) of 3rd corner
- (latitude, longitude) of 4th corner

The corner points should be entered clockwise, but the program can reverse a counterclockwise definition.

Imagery Sample Group: Each imagery sample group gives input for a single sampling technology. The first card image of the group should have a 16-character descriptive title and a value for the correlation of each pixel with the true value for the pixel. The next data cards define a translation table so that pixel values can be input as a character code instead of a numeric value for each pixel. The translate table consists of an integer, N, that gives the number of entries in the table, followed by character code 1, numeric value 1, character code 2, numeric value 2, ..., character code N, numeric value N. The image itself is entered a line at a time with each line described by:

- (latitude, longitude) of start of line
- (latitude, longitude) of end of line
- a character string of the indicated length giving values

#### Specific Format (Field Definitions)

Since data is entered in a fixed field format, it is important that all data appear in the correct card image columns. The following defines the format for each data group. The form of each data item is specified by showing the appropriate FORTRAN format description.

Note that all latitude values are entered as decimal degrees, positive for North latitude. All longitude values are entered as decimal degrees, positive for west of  $0^{\circ}$ . The current

version of the program remaps all data to a polar stereographic grid with standard latitude  $60^{\circ}$  N and standard longitude  $105^{\circ}$  W; this is appropriate for data in North America, but inadequate for data from other regions. The subroutine PCHRAP should be modified for other regions.

#### Comments and Title Groups

Anywhere in columns 2 through 80. Maximum of 50 comment lines.

#### Basin Boundary Group

Latitude: columns 2-9, F8.2

Longitude: columns 10-17, F8.2

Latitude-longitude pairs are entered one pair per card.

#### Point Data Group

Identifier: Columns 2-13, 3A4

Latitude: Columns 14-21, F8.2

Longitude: Columns 22-29, F8.2

Sample Value: Columns 30-37, F8.2

Correlation: Columns 38-44, F7.4

Decay Para.: Columns 45-51, F7.4

Each point is described by the information on a single card. If correlation or decay parameters are omitted, then the corresponding columns are left blank.

#### Line Data Group

Identifier: Columns 2-13, 3A4

Lat. of Start: Columns 14-21, F8.2

Long. of Start: Columns 22-29, F8.2

Lat. of end: Columns 30-37, F8.2

Long. of end: Columns 38-45, F8.2

Sample value: Columns 46-53, F8.2

Correlation: Columns 54-60, F7.4

Decay Para: Columns 61-67, F7.4

Each line is described by the information on a single card. Omitted values are noted by blanks in the appropriate columns.

#### Areal Sample Group

Descriptive Title: Columns 2-17, 4A4

Lat. of 1st corner: Columns 2-9, F8.2  
Long. of 1st corner: Columns 10-17, F8.2  
Lat. of 2nd corner: Columns 18-25, F8.2  
Long. of 2nd corner: Columns 26-33, F8.2  
Lat. of 3rd corner: Columns 34-41, F8.2  
Long. of 3rd corner: Columns 42-49, F8.2  
Lat. of 4th corner: Columns 50-57, F8.2  
Long. of 4th corner: Columns 58-65, F8.2  
Sample value: Columns 66-73, F8.2  
Correlation: Columns 74-80, F7.4

Each quadrilateral area is defined by the information on a single card. If the correlation field is left blank, then the value from the last card on which a correlation value was given is used. Each group can have only one title, but several areal sample groups can be included in a data set. Thus, if areal data from several technologies are available, data from each technology can be input as a separate group, with an identifying label.

#### Imagery Sample Group

Description Title: Columns 2-17  
Correlation Value: Columns 18-24, F7.4

Number of Values in Table : Columns 2-5, I4  
Character Code: Columns 6, A1  
Numeric Value: Columns 7-14, F8.2  
Charater Code: Column 15, A1  
Numeric Value: Columns 16-23, F8.2  
Character code: Column 24, A1  
Numeric Value: Columns 25-32, F8.2  
(Continue to Column 77, then begin new card with character code in Column 6)

Lat. of start: Columns 2-9, F8.2  
Long. of start: Columns 10-17, F8.2  
Lat. of end: Columns 18-25, F8.2  
Long. of end: Columns 26-33, F8.2  
Number of Values: Columns 34-37, I4  
Character string: Columns 38-80

Figure 3-1 gives a sample data set.

#### Program Output

The program produces two kinds of output: a graphic description of the distribution of sample values in the basin and a table that lists, for each sampling technology, the input measurement value, the computed weight assigned to that value, the correlation coefficient, the decay parameters and a symbol that relates this sample to the graphic display. The basin-wide areal average estimate is then given, along with the overall accuracy estimate. This estimate will be between 0.0 and 1.0, with a value of 1.0 indicating an exact estimate. The sample run in the next section shows this output.

\*  
THE DATA SET IS PROVIDED BY DR. E. L. PECK OF HYDEX.  
COTTONWOOD RIVER BASIN - AIRBORNE SOIL MOISTURE PROJECT  
THE BASIN HAS 9 VERTEX-POINTS.  
THERE ARE 6 POINT MEASUREMENTS, AND 16 LINE MEASUREMENTS.

\*

C

AIRBORNE SOIL MOISTURE PROJECT

B

44.295 94.445  
44.168 94.964  
43.952 95.369  
44.261 95.838  
44.200 95.917  
44.235 95.995  
44.430 95.817  
44.398 95.312  
44.459 95.291

N

P

POINT DATA02	44.296	94.973	27.0	0.94	0.17
POINT DATA04	44.472	95.488	23.0		
POINT DATA08	44.269	95.967	25.0		
POINT DATA09	44.298	95.648	28.0		
POINT DATA19	44.111	95.188	23.0		
POINT DATA23	44.308	94.504	26.0		

L

MN 501	44.235	95.881	44.235	95.959	26.6	0.85	0.17
MN 502	44.239	95.750	44.411	95.750	27.1		
MN 503C	44.244	95.610	44.244	95.755	31.5		
MN 504	44.160	95.619	44.228	95.619	29.8		
MN 505C	44.037	95.451	44.194	95.451	25.1		
MN 506	44.037	95.323	44.037	95.490	21.0		
MN 507	44.052	95.120	44.206	95.120	20.8		
MN 508C	44.239	95.145	44.406	95.145	29.9		
MN 509	44.393	95.114	44.393	95.412	24.6		
MN 510	44.231	95.271	44.231	95.474	27.1		
MN 511	44.231	95.130	44.231	95.255	28.7		
MN 512	44.243	94.990	44.243	95.125	22.0		
MN 513	44.287	94.870	44.246	94.990	29.2		
MN 514C	44.276	94.870	44.276	94.760	22.8		
MN 515	44.206	94.734	44.283	94.734	20.0		
MN 516	44.343	94.516	44.306	94.718	24.1		

FIGURE 3-1. A SAMPLE DATA SET

A

AREAL SAMPLE

44.500	96.000	44.500	95.250	44.250	95.250	44.250	96.000	24.000	0.65
44.500	95.250	44.50	94.500	44.250	94.500	44.250	95.250	26.000	
44.250	96.000	44.250	95.250	44.000	95.250	44.000	96.000	22.000	
44.250	95.250	44.250	94.500	44.000	94.500	44.000	95.250	23.000	

I

IMAGERY DATA 0.25

10A	22.0	B	22.50	C	23.0	D	23.5	E	24.0	F	24.5	G	25.0	H	25.5	
I	26.0	J	26.50													
43.750	96.000		44.500		96.000			4CDEF								
43.750	95.500		44.500		95.500			4CDEF								
43.750	95.000		44.500		95.000			4CDEF								
43.750	94.500		44.500		94.500			4CDEF								

E

S

FIGURE 3-1. A SAMPLE DATA SET. (Con't)

### Sample Run

The following sample run used the input file shown in Figure 3-1. Note that the text enclosed between asterisks in the input file is printed out when the input file is read. This allows an immediate check to be made as to whether the correct data file has been read.

RUN PCAMGRID

WELCOME TO CORRELATION AREA (GRID) METHOD PROGRAM

IN ORDER TO RUN THIS PROGRAM YOU MUST ALREADY HAVE AN EXISTING  
DATA FILE CONTAINING THE BASIN DATA, POINT, LINE AND AREAL SAMPLES

ENTER ' ' TO CONTINUE

INPUT DATA FILE NAME: XXXXXX.DAT  
SAMPLE.DAT

ENTER A TWO-DIGIT (01,05 - 20) OUTPUT UNIT NUMBER. 01 FOR USER TERMINAL  
01

THE DATA SET IS PROVIDED BY DR. E. L. PECK OF HYDEX.  
COTTONWOOD RIVER BASIN - AIRBORNE SOIL MOISTURE PROJECT  
THE BASIN HAS 9 VERTEX-POINTS.  
THERE ARE 6 POINT MEASUREMENTS, AND 16 LINE MEASUREMENTS.

WANT INPUT DATA TO BE PRINTED? ENTER Y OR N  
N

BASIN DATA IS READ:  
PROCESSING FOR BASIN DATA COMPLETED

BEGIN A NEW SAMPLING DATA GROUP:

POINT SAMPLING DATA IS READ:  
PROCESSING POINT DATA COMPLETED

LINE SAMPLING DATA IS READ:  
PROCESSING LINE DATA COMPLETED

AREAL SAMPLING DATA IS READ:  
PROCESSING AREAL DATA COMPLETED

IMAGERY SAMPLING DATA IS READ:  
PROCESSING IMAGERY DATA COMPLETED

PROCESSING FOR THIS DATA GROUP COMPLETED

AREAL SAMPLING SUMMARY:

SAMPLING TECH	CORRELATION	VALUE	WEIGHT
1	0.6417	23.81	0.8431
2	0.2500	24.00	0.1569

ENTER # OF GRID POINTS (INTEGER IN FOUR DIGITS)  
2500

## SAMPLING TECH DISTRIBUTION IN THE BASIN

# OF GRID POINTS ON THE BASIN = 1189  
THE WIDTH OF THE GRID = 0.3813

AIRBORNE SOIL MOISTURE PROJECT

SAMPLING TECH	MEASUREMENT VALUE	WEIGHT	CORRELATION	ALPHA	SYMBOL
A	23.8399	0.1993	0.680		(\$)
P	27.0000	0.0419	0.940	0.170	(A)
P	23.0000	0.0023	0.940	0.170	(B)
P	25.0000	0.0160	0.940	0.170	(C)
P	28.0000	0.0429	0.940	0.170	(D)
P	23.0000	0.0445	0.940	0.170	(E)
P	26.0000	0.0127	0.940	0.170	(F)
L	26.6000	0.0140	0.850	0.170	(1)
L	27.1000	0.0621	0.850	0.170	(2)
L	31.5000	0.0313	0.850	0.170	(3)
L	29.8000	0.0358	0.850	0.170	(4)
L	25.1000	0.0559	0.850	0.170	(5)
L	21.0000	0.0528	0.850	0.170	(6)
L	20.8000	0.0358	0.850	0.170	(7)
L	29.9000	0.0494	0.850	0.170	(8)
L	24.6000	0.0753	0.850	0.170	(9)
L	27.1000	0.0657	0.850	0.170	(0)
L	28.7000	0.0293	0.850	0.170	(#)
L	22.0000	0.0302	0.850	0.170	(&)
L	29.2000	0.0260	0.850	0.170	(+)
L	22.8000	0.0285	0.850	0.170	(-)
L	20.0000	0.0192	0.850	0.170	(=)
L	24.1000	0.0294	0.850	0.170	(?)

THE BASIN-WIDE AREAL AVERAGE ESTIMATE = 25.3654  
 THE OVERALL ACCURACY OF THIS ESTIMATE = 0.7518

ENTER ANOTHER # OF GRID POINTS (IF NO MORE, ENTER 0)  
 0

\*\*\* END \*\*\*

\*\*\*\* STOP

OK 11:37:48 11.848 4.360

NOTE: This run used 11.848 seconds of CPU time and 4.360 seconds of I/O time on a PRIME 750 computer (PRIMOS operating system).

## CHAPTER IV

### AIDS TO USERS

#### PROGRAM DESCRIPTION

This chapter describes several subtle features of the program. This is done solely for documentation reasons and is not meant to be read and understood by all users. The subtleties described here are important for two reasons: to simplify the overall program structure and to control the amount of storage needed for basin and sub-basin information.

The program is organized around a set of subroutines that process each type of input data group. As each data group is read, a subroutine that can process that type of data is called. This subroutine, with the help of others, stores the data in one large array, called the boundary data array. Initially this array is empty. As each group of data is read, new information is added to the array, or previous information is combined or compressed. By the end of a data set, the array is again empty. In the next subsection, the contents of the boundary data array is described. Flags that are used in processing the boundary data and information on how data in the array is combined or compressed are also explained in the following sections.

#### Boundary Data Array

The boundary data array is called BDCHAN and is used to store information about all of the boundaries currently being processed. There are several different types of boundaries. Each boundary type has certain important information that is relevant to it and must be stored. The elements of BDCHAN are of REAL type, and are defined as follows:

The first 7 relative positions of the boundary data array for each boundary hold various control information as

described below. The entire BDCHAN array is made up of multiple boundaries chained together.

<u>Relative Position</u>	<u>Identifier</u>	<u>Description</u>
1	TYPE	Boundary Type  = -1.0 to indicate a boundary which is to be deleted.  = 0.0 to indicate the original basin boundary. Only one type 0.0 boundary will exist.  = 1.0 to indicate a subbasin boundary. There may be multiple type 1.0 boundaries at any time.  = 2.0 to indicate a special "augmented" boundary. There can be 2 and only 2 type 2.0 boundaries at any time (if any).
2	NPTR	Pointer to next boundary chain in the BDCHAN array, i.e., BDCHAN (IFIX (Pointer value + 0.1)) is the type of the next boundary data chain. The pointer will be set to -1.0 for the last boundary data chain.
3	MPTS	Maximum number of vertex points which this boundary can contain.
4	NPTS	Actual number of vertex points which this boundary contains.
5	AREA	For Type = 0.0 or 1.0 boundaries only. Computed area in same units as x-y grid.
5	NINTER	For Type = 2.0 boundaries only. Number of intersection-type boundary points.
6	CORR	Computed correlation of areal average value to true average over this boundary. Set to -1.0 if not available. Not used for TYPE = 2.0.

7	VAL	Computed areal average value if available. Not defined if CORR = -1.0.
---	-----	--

The remaining relative positions hold the actual vertex points of the boundary. For TYPE = 0.0 or TYPE = 1.0 each vertex point is simply an x,y pair. For a type 2.0 boundary each vertex point has three values: x, y, and a "flag" which will be described in the next section. For type 0.0 and 1.0 boundaries, the remaining positions, relative to the start of this boundary data, are:

<u>Relative Position</u>	<u>Identifier</u>	<u>Description</u>
8	$x_1$	X position of first point.
9	$y_1$	Y position of first point.
10	$x_2$	X position of second point.
11	$y_2$	Y position of second point.
.	.	
.	.	
.	.	
$6+2*NPTS$	$x_{NPTS}$	X position of last point.
$7+2*NPTS$	$y_{NPTS}$	Y position of last point.

For boundaries with Type = 2.0, the remaining positions are:

<u>Relative Position</u>	<u>Identifier</u>	<u>Description</u>
8	$x_1$	X position of first point.
9	$y_1$	Y position of first point.
10	Flag <sub>1</sub>	Flag for first point (see below).

Note that the value of the pointer, NPTR, can be used to determine exactly how many locations are used for the description of each boundary.

11	$x_2$	X position of second point.
12	$y_2$	Y position of second point.

13	Flag <sub>2</sub>	Flag for second point.
.	.	
.	.	
.	.	
5+3*NPTS	X <sub>NPTS</sub>	X position of last vertex.
6+3*NPTS	Y <sub>NPTS</sub>	Y position of last vertex.
7+3*NPTS	Flag <sub>NPTS</sub>	Flag position of last vertex.

### Augmented Boundary Flags

The flags are used in two different ways: one when the augmented boundaries are in the process of being created, and a second way when they are completed. When the augmented sets are created, each is started with a list of original boundary points, then augmented with all points which are the intersection points of two boundaries. Each intersection point appears in both augmented boundaries. While the augmented sets are being created, these intersection points are simply numbered from 1 to NINTER (the number of intersections) and this numeric counter is used as the flag value for intersections points. After the augmented sets are completed, these intersection point flags are changed into pointers to the location (1...NPTS) of the associated intersection point in the other augmented boundary.

The flag value for the original points is simply a zero until the augmented sets are used to define subarea boundaries. During this process these flag values are changed from zero (not used) to a -1.0 to indicate that a point has been "used" in the subarea definition.

The flag values are summarized below:

#### DURING AUGMENTED BOUNDARY DEFINITION

<u>Flag Value</u>	<u>Description</u>
0.0	An original vertex point.
I	The I-th intersection point, $1 \leq I \leq NINTER$

#### DURING SUBBASIN DEFINITION

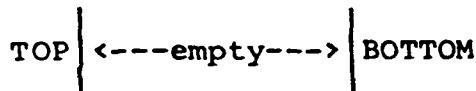
-1.0	An original point which has been used.
0.0	An original point which has not been used

I.(>0) A pointer to the I-th point (1  $\leq$  I  $\leq$  NPTS) in the other augmented boundary set which is the same intersection point as this one. For example, If Flag<sub>10</sub> = 12 then the value of Flag<sub>12</sub> in the other augmented set is 10.

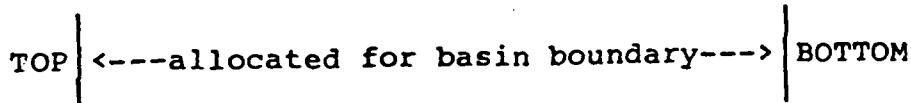
#### Boundary Data Chains

As each group of data is read, it is processed and stored in the array BDCHAN. The resulting "chain" of boundary data can have one of six forms:

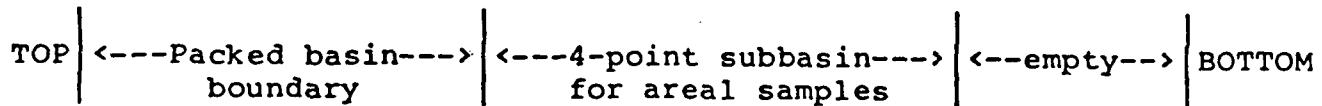
(1) Empty:



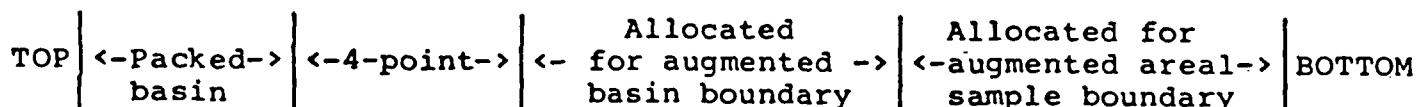
(2) Basin only:



(3) Packed basin plus areal sample:



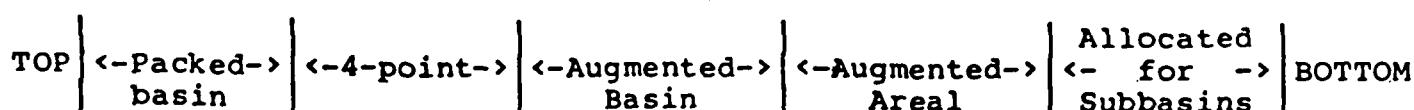
(4) Allocated for augmented boundaries:



(5) Packed Augmented Boundaries:



(6) Subbasin Boundaries:



The program changes the status of the boundary data chains in the following way:

The program starts with BDCHAN in status (1) and returns to status (1) whenever a (B)asin control card is encountered. After a B control card is encountered, the basin data is used by subroutine PCBALC to allocate the entire chain to a type 0 basin boundary. BDCHAN now has form (2). After the basin boundary points have all been processed, PCBDPK packs the basin boundary. After packing, a 4-point type 1 boundary is set aside for any areal samples. This puts BDCHAN into form (3).

As each areal sample is processed, its boundary is placed in the 4-point type 1 area. The empty space below this is allocated equally for the type 2 boundaries: The augmented basin boundary and the augmented areal sample boundary. This allocation requires two calls to PCBALC. This process gives status (4), ready for a call to subroutine PCBAUG.

After the augmented boundaries are defined by PCBAUG, the array is packed into form (5) by PCBDPK.

All of the remaining empty space is set aside for a single type 1 boundary to hold the subbasins as created by calls to PCSUBB. This is form 6.

When all sub-basins have been found, then the augmented basin, augmented areal, and the allocation for sub-basin will all be set to type = -1 and deleted via PCBDPK. This restores status 3 for the next areal sample.

#### Input Data Processing

The following is a list of the subroutines that process the input data, with a brief description of what each does. Note

that the main purpose of these routines is to load data into the boundary data array BDCHAN. The variable NCHAN gives the number of elements currently in BDCHAN.

Subroutine Name

PCINST	Comment Group - just echo prints until next * card found.
PCINC	Control Group.
PCINB	Basin Boundary Group. <ul style="list-style-type: none"><li>- set NCHAN=0</li><li>- allocate type 0 boundary</li><li>- read boundary points</li><li>- convert boundary from (lat, Long) to (x,y)</li><li>- pack BDCHAN</li><li>- allocate 4-point type 1 boundary</li><li>- compute basin area (assures clockwise order too).</li></ul>
PCINN	New Data Case <ul style="list-style-type: none"><li>- check for valid type 0 boundary</li><li>- read data and output options</li><li>- initialize NPDTA, NLDTA, and NADTA = 0</li><li>- set areal average and computed correlation of basin boundary to -1.0</li><li>- set code indicating that data input is valid, i.e., that PCINP, PCINL, PCINA, and PCINI can be called.</li></ul>
PCINP	Point Data Group <ul style="list-style-type: none"><li>- read each value into common block PCPDTA and convert (lat, long) to (x,y).</li></ul>
PCINL	Line Data Group <ul style="list-style-type: none"><li>- read flight lines into common block PCLDTA and convert (lat, long) to (x,y).</li></ul>
PCINA	Areal Data Group <ul style="list-style-type: none"><li>- This is a rather complex routine. Basically, it combines all the areal sample values from a single technology into one entry in common block PCADTA.</li></ul>

PCINI	Image Data Group - Combines all values from an imagery type technology into a single entry in common.
PCINE	End of Data Case - Perform in order:--combine all existing areal and image technologies in common block PCATA into single basin estimate (See Subroutine PCCOMB). -- use grid method to combine all sampling technologies. See PCGRID. -- output results. -- set code indicating that data input is <u>not</u> valid until after "N" card.
PCINS	Stop Program (also process like end of data if sample values available).

#### ACKNOWLEDGEMENTS

The assistance and leadership provided by Dr. Thomas Schmugge of the National Aeronautics and Space Administration and Dr. Al Rango (Formerly of NASA and now with the ARS Department of Agriculture) is greatfully acknowledged.

E. R. Johnson (formerly a consultant with Hydex) now with the Hydrologic Research Laboratory, National Weather Service, NOAA, Silver Spring, Maryland, designed the CAM program. Actual implementation of the CAM computer program was done by M. Y. Wong, Automated Sciences Group, Inc., Silver Spring, Maryland, under subcontract from Hydex Corporation.

## REFERENCES

1. E. L. Peck, R. S. McQuivey, T. N. Keefer, E. R. Johnson and J. L. Erikson, "Review of Hydrologic Models for Evaluating Use of Remote Sensing Capabilities," NASA CR166674, Goddard Space Flight Center, Greenbelt, Maryland, March 1981, pp.99.
2. E. L. Peck, T. N. Keefer, and E. R. Johnson, "Strategies for Using Remotely Sensed Data in Hydrologic Models," NASA CR166729, Goddard Space Flight Center, Greenbelt, Maryland, July 1981, pp.84.
3. E. R. Johnson, E. L. Peck, and T. N. Keefer, "Combining Remotely Sensed and Other Measurements for Hydrologic Areal Averages", NASA CR170457, Goddard Space Flight Center, Greenbelt, Maryland, October 1982, pp.164.
4. E. L. Peck, E. R. Johnson and T. N. Keefer, "Creating A Bridge Between Remote Sensing and Hydrologic Models," NASA CR170517, Goddard Space Flight Center, Greenbelt, Maryland, January 31, 1983, pp.114.
5. T. R. Carroll, J. E. Glynn, and B. E. Goodison, "A Comparison of U. S. and Canadian Airborne Gamma Radiation Snow Water Equivalent Measurements. Proc. 51st Annual Western Snow Conference, Vancouver, Washington, U.S.A., 1983, pp.27-37.
6. T. J. Schmugge, "Microwave Approaches in Hydrology. Potogramm. Engineering and Remote Sensing 46 (4), 1980, pp.495-507.

APPENDIX A

CAM PROGRAM LISTING

```

PROGRAM PCAMGRID
C
C CORRELATION AREA GRID METHOD
C   THE PURPOSE OF THIS PROGRAM IS TO OBTAIN A ESTIMATE AREAL AVERAGE
C   VALUE FOR A BOUNDED BASIN BASED ON MEASUREMENTS OBTAINED FROM
C   POINT SAMPLES, LINE SAMPLES, OR AREAL SAMPLES
C
C
      COMMON /PCPDTA/MPDTA,NPDTA,ALPHAP(500),CP(500),XPDTA(500),
      *           YPDTA(500),CAPDTA(500),VALP(500),DESCRP(3,500)
      COMMON /PCLDTA/MLDTA,NLDTA,ALPHAL(100),DL(100),XLDTA(2,100),
      *           YLDTA(2,100),CALDTA(100),VALL(100),DESCRL(3,100)
      COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
      *           DESCRA(4,20)
      COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
      CHARACTER CODE*1,ANS1*1,FNAME*10,TITLE*79,INPLST*1
C
      IOUNIT=1
      WRITE(1,1)
1 FORMAT(//1X,'WELCOME TO CORRELATION AREA (GRID) METHOD PROGRAM',
      *           //1X,'IN ORDER TO RUN THIS PROGRAM YOU MUST ALREADY ',
      *           'HAVE AN EXISTING',/1X,'DATA FILE CONTAINING THE BASIN ',
      *           'DATA, POINT, LINE AND AREAL SAMPLES')
10 WRITE(1,3)
3 FORMAT(/1X,'ENTER ' ' ' TO CONTINUE')
C
      READ(1,5) ANS1
5 FORMAT(A1)
      IF(ANS1 .NE. ' ') GO TO 9990
C
40 WRITE(1,50)
50 FORMAT(/1X,'INPUT DATA FILE NAME: XXXXXX.DAT')
      READ(1,55) FNAME
55 FORMAT(A10)
C
C
      OPEN(UNIT=29,FILE=FNAME,STATUS='OLD',ACCESS='SEQUENTIAL',
      *           FORM='FORMATTED',ERR=10,IOSTAT=IOVAL)
C
      IF(IOVAL .EQ. 0) GO TO 70
      WRITE(1,60)
60 FORMAT(/1X,'THERE IS A PROBLEM WITH THE DATA FILE, REENTER')
      GO TO 10
C
70 WRITE(1,75)
75 FORMAT(/1X,'ENTER A TWO-DIGIT (01,05 - 20) OUTPUT UNIT NUMBER.',
      *           ' 01 FOR USER TERMINAL ')
      READ(1,78) IOUNIT
78 FORMAT(I2)
C
100 READ(29,5) CODE
150 IF(CODE .NE. '*') GO TO 200
      CALL PCINST(IOUNIT,*9990)
      GO TO 100

```

C  
200 IF(CODE .NE. 'C') GO TO 300  
CALL PCINC(CODE,TITLE,INPLST)  
GO TO 150

C  
300 IF(CODE .NE. 'B') GO TO 400  
WRITE(IOUNIT,310)  
310 FORMAT(/1X,'BASIN DATA IS READ:')  
CALL PCINB(CODE,IOUNIT,INPLST,\*9990)  
WRITE(IOUNIT,320)  
320 FORMAT(1X,'PROCESSING FOR BASIN DATA COMPLETED')  
GO TO 150

C  
400 IF(CODE .NE. 'N') GO TO 500  
WRITE(IOUNIT,410)  
410 FORMAT(/1X,'BEGIN A NEW SAMPLING DATA GROUP:')  
CALL PCINN(CODE,IOUNIT,INPLST,\*9990)  
WRITE(IOUNIT,420)  
420 FORMAT(/1X,'PROCESSING FOR THIS DATA GROUP COMPLETED')  
GO TO 150

C  
500 IF(CODE .NE. 'E') GO TO 600  
CALL PCINE(CODE,TITLE,IOUNIT,\*9990)  
GO TO 150

C  
600 IF(CODE .NE. 'S') GO TO 1000  
CALL PCINS(IOUNIT)  
GO TO 9990

C  
1000 WRITE(IOUNIT,1010)  
1010 FORMAT(/1X,'ERROR IN CONTROL CODE')  
C  
9990 CONTINUE  
C  
STOP  
END

```
BLOCK DATA
COMMON /PCPDTA/MPDTA,NPDTA,ALPHAP(500),CP(500),XPDTA(500),
*           YPDTA(500),CAPDTA(500),VALP(500),DESCRP(3,500)
COMMON /PCLDTA/MLDTA,NLDTA,ALPHAL(100),CL(100),XLDTA(2,100),
*           YLDTA(2,100),CALDTA(100),VALL(100),DESCRL(3,100)
COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
*           DESCRA(4,20)
COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
```

C

C

```
DATA MCHAN/10000/
DATA MPDTA/500/
DATA MLDTA/100/
DATA MADTA/2000/
DATA MATDTA/20/
DATA NGRID/400/
```

C

```
END
```

```

C *** SUBROUTINE PCHRAP ***
C
C OBJECTIVES:
C   CONVERTS THE (LATITUDE,LONGITUDE) LOCATION ON EARTH TO THE (X,Y)
C   HRAP GRIP SYSTEM LOCATION. (ASSUMES NORTH POLE = (401,1601) ).  

C
C
C   SUBROUTINE PCHRAP(NPAIR,FLAT,FLONG,XYGRID)
C
C   DIMENSION FLAT(NPAIR),FLONG(NPAIR),XYGRID(2,NPAIR)
C
C   DATA DEGRAD/.01745329/,EARTH/6371.2/,STLON/105./,STLAT/60./
C
C   XMESH=4.7625
C   TLAT=STLAT*DEGRAD
C   RE=(EARTH*(1.+SIN(TLAT)))/XMESH
C
C   DO 20 I=1,NPAIR
C     XLAT=FLAT(I)*DEGRAD
C     WLONG=(FLONG(I)+180.-STLON)*DEGRAD
C     R=(RE*COS(XLAT))/(1.+SIN(XLAT))
C     XYGRID(1,I)=R*SIN(WLONG)+401.
C     XYGRID(2,I)=R*COS(WLONG)+1601.
20   CONTINUE
C
C   RETURN
C   END

```

```

C *** SUBROUTINE PCAREA ***
C
C OBJECTIVES:
C   1. FINDS THE AREA OF A GIVEN BOUNDARY
C   2. ASSURES THAT THE BOUNDARY IS DEFINED IN A CLOCKWISE FASHION.
C
C PRINCIPAL VARIABLES:
C
C   NAME  I/O  DIMENSION  DESCRIPTION
C   ----  ---  -----  -----
C   BND    I    (NPV,N)  VERTICES OF BOUNDARY
C           (O)          BND(1,I) = X-LOCATION OF THE I-TH VERTEX
C                           BND(2,I) = Y-LOCATION OF THE I-TH VERTEX
C                           ORDER OF PTS IS REVERSED ON OUTPUT IF ISW=1
C   NPV    I    -        NUMBER OF DATA VALUES PER VERTEX (=2 OR 3)
C   N      I    -        NUMBER OF VERTICES IN BOUNDARY
C   AREA   O    -        COMPUTED AREA
C   ISW    O    -        RETURN STATUS, =0  NORMAL RETURN
C                           =1  PTS RETURNED IN REVERSED ORDER
C                           ==-1  ERROR OCCURED
C
C
C   SUBROUTINE PCAREA(BND,N,NPV,AREA,ISW)
C
C   DOUBLE PRECISION DAREA
C   DIMENSION BND(NPV,N)
C
C   DAREA=0.
C
C TEST ARGUMENTS
C
C   ISW=-1
C   IF(N .LT. 3 .OR. NPV .LT. 2) RETURN
C   ISW=0
C
C FIND THE AREA
C
C   DO 100 I=1,N-1
C       AD=BND(1,I+1)*BND(2,I)-BND(1,I)*BND(2,I+1)
C       DAREA=DAREA+AD
C 100 CONTINUE
C   DAREA=DAREA+BND(1,1)*BND(2,N)-BND(1,N)*BND(2,1)
C   AREA=DAREA/2.
C
C   IF(AREA .GE. 0.0) RETURN
C   AREA=ABS(AREA)
C
C THE BOUNDARY IS IN COUNTERCLOCKWISE, ORDER OF POINTS WILL BE REVERSED
C
C   ISW=1
C   NHALF=IFIX(N/2+0.1)
C   DO 300 IUP=1,NHALF
C       IDOWN=N-IUP+1
C       DO 200 I=1,NPV
C           TMP=BND(I,IUP)
C           BND(I,IUP)=BND(I,DOWN)
C           BND(I,DOWN)=TMP
C 200   CONTINUE
C 300   CONTINUE

```

200 CONTINUE  
300 CONTINUE  
C  
RETURN  
END

```

C *** SUBROUTINE PCIFIN ***
C
C OBJECTIVES:
C   DETERMINES IF A GIVEN POINT INSIDE OR OUTSIDE A BOUNDARY
C
C PRINCIPAL VARIABLES:
C   NAME  I/O  DIMENSION  DESCRIPTION
C   ----  ---  -----  -----
C   X      I      -      X-POSITION OF POINT TO BE TESTED
C   Y      I      -      Y-POSITION OF POINT TO BE TESTED
C   BND    I      (NPV,N)  VERTICES OF BOUNDARY
C                   BND(1,I) = X-POSITION OF THE I-TH VERTEX
C                   BND(2,I) = Y-POSITION OF THE I-TH VERTEX
C   NPV    I      -      NUMBER OF DATA VALUES PER VERTEX (=2 OR 3)
C   N      I      -      NUMBER OF VERTICES IN BOUNDARY
C   INOUT  O      -      =0 FOR (X,Y) OUTSIDE OF BOUNDARY
C                   =1 FOR (X,Y) INSIDE BOUNDARY
C
C
C   SUBROUTINE PCIFIN(X,Y,BND,N,NPV,INOUT)
C
C   DIMENSION BND(NPV,N)
C
C INSIDE OR OUTSIDE OF A BOUNDARY DEPENDS ON THE NO. OF CROSSING OF
C ANY HALF-LINE FROM (X,Y) TO THE BOUNDARY. ODD-INSIDE, EVEN-OUTSIDE.
C
C   NCROS=0
C
C   DO 100 I=1,N
C     X1=BND(1,I)
C     Y1=BND(2,I)
C     I2=I+1
C     IF(I2 .GT. N) I2=1
C     X2=BND(1,I2)
C     Y2=BND(2,I2)
C
C   END POINTS OF BOUNDARY LINE FOUND: (X1,Y1),(X2,Y2)
C
C   IF(X .LT. X1 .AND. X .LT. X2) GO TO 100
C   IF(X .GT. X1 .AND. X .GT. X2) GO TO 100
C   IF(Y .LT. Y1 .AND. Y .LT. Y2) GO TO 100
C
C   TEST X1=X2 (VERTICAL LINE) OR NOT
C
C   IF(X1 .EQ. X2) GO TO 50
C   YSTAR=(X*(Y1-Y2)+(X1*Y2-Y1*X2))/(X1-X2)
C   IF (YSTAR .LE. Y) NCROS=NCROS+1
C   GO TO 100
C 50  IF(Y .LE. Y1 .OR. Y .LE. Y2) NCROS=NCROS+1
C 100 CONTINUE
C
C   INOUT=MOD(NCROS,2)
C
C   RETURN
C   END

```

```

C *** SUBROUTINE PCBALC ***
C
C OBJECTIVES:
C   ALLOCATES SPACE FOR A NEW BOUNDARY TO BDCHAN ARRAY IN COMMON BLOCK PCBDCH
C
C PRINCIPAL VARIABLES:
C   NAME  I/O  DIMENSION  DESCRIPTION
C   ----  ---  -----  -----
C   ITYP   I      -      BOUNDARY TYPE TO BE ALLOCATED
C                   =0 FOR ORIGINAL BASIN BOUNDARY
C                   =1 FOR A SUB-BASIN BOUNDARY
C                   =2 FOR AN AUGMENTED BOUNDARY
C   MPTS   I/O     -      ON INPUT: # OF VERTEX POINTS TO ALLOCATED
C                           SET TO -1 TO ALLOCATE ALL AVAILABLE SPACE
C                           OUTPUT: ACTUAL # OF VERTIEX POINTS ALLOCATED
C   LOC    O      -      LOCATION OF BOUNDARY IN BDCHAN ARRAY
C                           SET LOC=-1 IF NO SPACE COULD BE ALLOCATED
C
C
C   SUBROUTINE PCBALC(ITYP,MPTS,LOC)
C
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
C   LOC=-1
C   NOPEN=MCHAN-NCHAN
C   NPV=2
C   IF(ITYP .EQ. 2) NPV=3
C   MAX=(NOPEN-7)/NPV
C
C   MAX IS THE MAX # OF VERTIEX POINTS CAN BE ALLOCATED IN BDCHAN FOR THIS
C   BOUNDARY. NOTE THAT ALL BOUNDARY MUST HAVE THREE OR MORE POINTS
C
C   IF(MAX .LT. 3) RETURN
C   IF(MPTS .EQ. -1) MPTS=MAX
C   IF(MPTS .GT. MAX) MPTS=MAX
C   LOC=NCHAN+1
C
C   LOC IS THE FIRST AVAILABLE APACE FOR THE NEW BOUNDARY
C
C   IF(NCHAN .EQ. 0) GO TO 100
C
C   FIND LAST EXISTING BOUNDARY, SET ITS POINTER TO LOC
C
C   NPTR=1
50  LLAST=NPTR
    NPTR=IFIX(BDCHAN(LLAST+1))
    IF(NPTR .GT. 0) GO TO 50
    BDCHAN(LLAST+1)=FLOAT(LOC)+0.1
C
C   FILL VALUES FOR THIS BOUNDARY, THE SECOND POSITION IS -1 SINCE IT IS THE
C   LAST
C   THE 4-TH POSITION WILL BE FILLED WHEN BOUNDARY POINTS IS READ AND SUBROUTINE
C   PCBDPK IS CALLED

```

```
C
100  BDCHAN(LOC)=FLOAT(ITYP)+0.1
      BDCHAN(LOC+1)=-1.0
      BDCHAN(LOC+2)=FLOAT(MPTS)+0.1
      BDCHAN(LOC+3)=0.0
      BDCHAN(LOC+4)=0.0
      BDCHAN(LOC+5)=-1.0
      BDCHAN(LOC+6)=0.0
C
      IF(ITYP .EQ. 2) NCHAN=NCHAN+MPTS*3+7
      IF(ITYP .NE. 2) NCHAN=NCHAN+MPTS*2+7
C
      RETURN
      END
```

```

C *** SUBROUTINE PCBDPK ***
C
C OBJECTIVES:
C   PACKS THE BOUNDARY DATA CHAINS OF ARRAY BDCHAN IN COMMON BLOCK PCBDCH
C   LEAVING ALL AVAILABLE SPACE AT THE END.  BOUNDARIES WHICH HAVE TYPE=-1.
C   ARE DELETED AS PART OF THE PACKING PROCESS.  THE VALUE OF NCHAN IS
C   RECOMPUTED AS WELL.
C
C PRINCIPAL VARIABLES:
C   NAME  I/O  DIMENSION  DESCRIPTION
C   ----  ---  -----  -----
C   MCHAN  I      -      DIMENSION OF ARRAY BDCHAN
C   NCHAN  I/O     -      ACTUAL LENGTH USED IN BDCHAN
C   BDCHAN I      MCHAN  BOUNDARY DATA ARRAY
C
C
C   SUBROUTINE PCBDPK
C
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
C   IF(IFIX(BDCHAN(1)) .EQ. -1 .AND. BDCHAN(2) .LT. 0.0) GO TO 140
C
C   NPTR=1
C   IF(NCHAN .LE. 0) GO TO 140
C
C TEST NEXT BOUNDARY FOR EMPTY SPACE
C
C 20 LOC1=NPTR
C   IF(LOC1 .LT. 1) GO TO 100
C   NPTR=IFIX(BDCHAN(LOC1+1))
C   MPTS=IFIX(BDCHAN(LOC1+2))
C   NPTS=IFIX(BDCHAN(LOC1+3))
C   ITYP=IFIX(BDCHAN(LOC1))
C   IF(ITYP .LT. 0) MOVE=NPTR-LOC1
C   IF(ITYP .EQ. 0 .OR. ITYP .EQ. 1) MOVE=2*(MPTS-NPTS)
C   IF(ITYP .EQ. 2) MOVE=3*(MPTS-NPTS)
C   IF(MOVE .LE. 0) GO TO 20
C
C SCROLL UP MOVE WORDS
C FIRST MUST FIX ALL POINTERS
C
C   LOC2=LOC1
C 50 IF(LOC2 .LT. 1) GO TO 55
C   N=IFIX(BDCHAN(LOC2+1))
C   BDCHAN(LOC2+1)=FLOAT(IFIX(BDCHAN(LOC2+1))-MOVE)+0.1
C   IF(IFIX(BDCHAN(LOC2+1)) .LT. -1) BDCHAN(LOC2+1)=-1.0
C   LOC2=N
C   GO TO 50
C
C 55 BDCHAN(LOC1+2)=BDCHAN(LOC1+3)
C   N1=LOC1+2*MPTS+7-MOVE
C   IF(ITYP .EQ. 2) N1=LOC1+3*MPTS+7-MOVE
C   IF(ITYP .EQ. -1) N1=LOC1
C   DO 60 I=N1,NCHAN-MOVE
C     BDCHAN(I)=BDCHAN(I+MOVE)

```

```
60 CONTINUE
NCHAN=NCHAN-MOVE
NPTR=IFIX(BDCHAN(LOC1+1))
IF(ITYP .EQ. -1) NPTR=LOC1
GO TO 20
C
100 IF(NCHAN .LE. 0) GO TO 140
NPTR=1
120 LOC1=NPTR
NPTR=IFIX(BDCHAN(LOC1+1))
IF(NPTR .GT. 0) GO TO 120
MPTS=IFIX(BDCHAN(LOC1+2))
NCHAN=LOC1+6+2*MPTS
IF(IFIX(BDCHAN(LOC1)) .EQ. 2) NCHAN=NCHAN+MPTS
IF(ITYP .LT. 0) NCHAN=LOC1-1
RETURN
C
C
140 NCHAN=0
RETURN
END
```

```
C *** SUBROUTINE PCINST ***
C
C OBJECTIVES:
C   COMMENT GROUP
C   JUST ECHO AND PRINT ALL COMMENTS UNTIL NEXT * FOUND
C
C
C   SUBROUTINE PCINST(IONUNIT,*)
C
C   CHARACTER COMMNT*79, CODE*1
C
C   1 FORMAT(A1,A79)
C   2 FORMAT(1X,A79)
C   3 FORMAT(/1X,'TOO MANY COMMENT LINES (MAX: 50 LINES)')
C
C   I=0
100 I=I+1
      READ(29,1) CODE,COMMNT
      IF(CODE .EQ. '*') RETURN
      WRITE(IONUNIT,2) COMMNT
      IF(I .GT. 50) GO TO 9000
C
      GO TO 100
C
9000 WRITE(IONUNIT,3)
C
      RETURN1
      END
```

```

C *** SUBROUTINE PCINB ***
C
C OBJECTIVES:
C   BASIN BOUNDARY GROUP
C     1. SETS NCHAN=0
C     2. ALLOCATES TYPE 0 BOUNDARY
C     3. READS BASIN BOUNDARY POINTS
C     4. CONVERTS BOUNDARY FROM (LAT.,LONG.) TO (X,Y) HRAP SYSTEM
C     5. PACKS BOUNDARY POINTS IN BDCHAN
C     6. COMPUTES BASIN AREA
C     7. ALLOCATES 4-POINT TYPE 1 BOUNDARY
C
C
C   SUBROUTINE PCINB(CODE, IOUNIT, INPLST, *)
C
C   COMMON /PCBDCH/MCHAN, NCHAN, BDCHAN(10000)
C
C   DIMENSION FLAT(1000), FLONG(1000), BXY(2,1000)
C
C
C   CHARACTER CODE*1, INPLST*1
C
C   ALLOCATE TYPE 0 BOUNDARY
C
C     NCHAN=0                      /* BDCHAN IS EMPTY */
C     MPTS1=-1
C     CALL PCBALC(0, MPTS1, LOC1)
C
C     NPTS=0
C     DO 100 I=1, MPTS1+1
C       READ(29,1) CODE, FLAT(I), FLONG(I)
C       IF(CODE .NE. ' ') GO TO 500
C       NPTS=NPTS+1
C 100 CONTINUE
C     IF(NPTS .GT. MPTS1) GO TO 9000
C
C 500 CALL PCHRAP(NPTS, FLAT, FLONG, BXY)
C
C     CALL PCAREA(BXY, NPTS, 2, AREA, ISW)
C     IF(ISW .EQ. -1) GO TO 9000
C
C     DO 1000 I=1, NPTS
C       BDCHAN(LOC1+5+2*I)=BXY(1,I)
C       BDCHAN(LOC1+6+2*I)=BXY(2,I)
C
C     IF(INPLST .EQ. 'N') GO TO 1000
C     WRITE(IOUNIT,2) FLAT(I), FLONG(I), BXY(1,I), BXY(2,I)
C
C 1000 CONTINUE
C
C     BDCHAN(LOC1+4)=AREA
C     BDCHAN(LOC1+3)=FLOAT(NPTS)+0.1
C
C     CALL PCBDPK
C
C   ALLOCATE 4-POINT TYPE 1 BOUNDARY

```

```
C
MPTS2=4
CALL PCBALC(1,MPTS2,LOC2)
C
RETURN
C
C ERROR
C
9000 WRITE(1,9100) MPTS
9100 FORMAT(/1X,'ERROR: THERE IS A PROBLEM WITH THE BASIN BOUNDARY',
*           /1X,'      # OF BOUNDARY PTS SHOULD BETWEEN 3 AND ',I3)
C
RETURN1
C
1 FORMAT(A1,2F8.2)
2 FORMAT(1X,2F10.3,10X,(' ',2F12.3,''))
C
END
```

```
C *** SUBROUTINE PCINC ***
C
C OBJECTIVES:
C   CONTROL GROUP
C     1. I/O OPTIONS
C     2. RUN TITLES
C     3. DEBUG FLAGS
C
C
C   SUBROUTINE PCINC(CODE,TITLE,INPLST)
C
C   CHARACTER TITLE*79, CODE*1, INPLST*1
C
C   READ(29,1) CODE, TITLE
C
C   I/O OPTIONS
C
C   WRITE(1,5)
C   READ(1,2) INPLST
C
C   IF(CODE .NE. ' ') RETURN
C   READ(29,2) CODE
C   RETURN
C
C   I/O OPTIONS
C   DEBUG FLAGS
C
C   1 FORMAT(A1,A79)
C   2 FORMAT(A1)
C   5 FORMAT(/1X,'WANT INPUT DATA TO BE PRINTED? ENTER Y OR N')
C
C   END
```

```

C *** SUBROUTINE PCINN ***
C
C OBJECTIVES:
C   NEW DATA CASE
C     1. CHCEK FOR VALID TYPE 0 BOUNDARY
C     2. READ DATA AND OUTPUT OPTION
C     3. INITIALIZE NPDTA,NLDTA,NADTA, AND NATDTA
C     4. SET AREAL AVERAGE & COMPUTED CORRELATION OF BASIN BOUNDARY TO -1.
C     5. SET CODE INDICATING THAT DATA INPUT IS VALID,
C        I.E., PCINP, PCINL, PCINA, AND PCINI CAN BE CALLED
C
C
C SUBROUTINE PCINN(CODE,IOUNIT,INPLST,*)
C
C   COMMON /PCPDTA/MPDTA,NPDTA,ALPHAP(500),CP(500),XPDTA(500),
C   *           YPDTA(500),CAPDTA(500),VALP(500),DESCRP(3,500)
C   COMMON /PCLDTA/MLDTA,NLDTA,ALPHAL(100),CL(100),XLDTA(2,100),
C   *           YLDTA(2,100),CALDTA(100),VALL(100),DESCRL(3,100)
C   COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
C   *           DESCRA(4,20)
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
C   CHARACTER CODE*1,INPLST*1
C
C CHECK FOR VALID TYPE 0 BOUNDARY
C
C   IF(IFIX(BDCHAN(1)) .NE. 0) GO TO 9000
C   IF(IFIX(BDCHAN(4)) .EQ. 0) GO TO 9100
C
C READ DATA AND OUTPUT OPTIONS
C
C
C
C INITIALIZE
C
C   NPDTA=0
C   NLDTA=0
C   NATDTA=0
C
C SET AREAL AVERAGE AND COMPUTED CORRELATION OF BASIN BOUNDARY TO -1.0
C
C   BDCHAN(6)=-1.0
C   BDCHAN(7)=-1.0
C
C SET CODE INDICATING THAT DATA INPUT IS VALID
C
C   READ(29,2) CODE
C   2 FORMAT(A1)
1000 IF(CODE .EQ. 'P') GO TO 1100
    IF(CODE .EQ. 'L') GO TO 1200
    IF(CODE .EQ. 'A') GO TO 1300
    IF(CODE .EQ. 'I') GO TO 1400
    RETURN

```

```

C
1100 WRITE(IONUNIT,1110)
1110 FORMAT(/1X,'POINT SAMPLING DATA IS READ:')
CALL PCINP(CODE,IONUNIT,INPLST)
WRITE(IONUNIT,1120)
1120 FORMAT(1X,'PROCESSING POINT DATA COMPLETED')
GO TO 1000

C
1200 WRITE(IONUNIT,1210)
1210 FORMAT(/1X,'LINE SAMPLING DATA IS READ:')
CALL PCINL(CODE,IONUNIT,INPLST)
WRITE(IONUNIT,1220)
1220 FORMAT(1X,'PROCESSING LINE DATA COMPLETED')
GO TO 1000

C
1300 WRITE(IONUNIT,1310)
1310 FORMAT(/1X,'AREAL SAMPLING DATA IS READ:')
CALL PCINA(CODE,IONUNIT,INPLST,*9200)
WRITE(IONUNIT,1320)
1320 FORMAT(1X,'PROCESSING AREAL DATA COMPLETED')
GO TO 1000

C
1400 WRITE(IONUNIT,1410)
1410 FORMAT(/1X,'IMAGERY SAMPLING DATA IS READ:')
CALL PCINI(CODE,IONUNIT,INPLST,*9200)
WRITE(IONUNIT,1420)
1420 FORMAT(1X,'PROCESSING IMAGERY DATA COMPLETED')
GO TO 1000

C
9000 WRITE(IONUNIT,9010)
9010 FORMAT(/1X,'ERROR: BASIN BOUNDARY IS NOT OF TYPE 0')
RETURN1
9100 WRITE(IONUNIT,9110)
9110 FORMAT(/1X,'ERROR: THERE IS NO BASIN BOUNDARY POINTS')
RETURN1
9200 CONTINUE
RETURN1

C
      END

```

```

C *** SUBROUTINE PCINP ***

C
C OBJECTIVES:
C   POINT DATA GROUP
C     1. READ POINT DATA INTO COMMON BLOCK PCPDTA
C     2. CONVERT (LAT.,LONG.) TO (X,Y) HRAP COORDINATE SYSTEMS
C
C
C   SUBROUTINE PCINP(CODE,IOUNIT,INPLST)
C
C   COMMON /PCPDTA/MPDTA,NPDTA,ALPHAP(500),CP(500),XPDTA(500),
C   *           YPDTA(500),CAPDTA(500),VALP(500),DESCRP(3,500)
C
C   DIMENSION PXY(2,500)
C
C
C   CHARACTER CODE*1,INPLST*1
C
C   DO 100 I=1,MPDTA+1
C     READ(29,1) CODE,(DESCRP(J,I),J=1,3),XPDTA(I),YPDTA(I),VALP(I),
C   *           CP(I),ALPHAP(I)
C     IF(CODE .NE. ' ') GO TO 500
C     IF(I .GT. 1 .AND. CP(I) .LT. .00001) CP(I)=CP(I-1)
C     IF(I .GT. 1 .AND. ALPHAP(I) .LT. .00001) ALPHAP(I)=ALPHAP(I-1)
C     NPDTA=NPDTA+1
C 100 CONTINUE
C     IF(NPDTA .GT. MPDTA) GO TO 9000
C
C
C   500 CALL PCHRAP(NPDTA,XPDTA,YPDTA,PXY)
C
C   DO 600 I=1,NPDTA
C     IF(INPLST .EQ. 'N') GO TO 550
C     WRITE(IOUNIT,2) (DESCRP(J,I),J=1,3),XPDTA(I),YPDTA(I),VALP(I),
C   *           CP(I),ALPHAP(I),PXY(1,I),PXY(2,I)
C   550 XPDTA(I)=PXY(1,I)
C           YPDTA(I)=PXY(2,I)
C   600 CONTINUE
C
C   RETURN
C
C-ERROR: TOO MANY POINT-SAMPLES
C
C   9000 WRITE(IOUNIT,9100) NPDTA
C   9100 FORMAT(/1X,'ERROR: # OF POINT-AMPLES CAN NOT EXCEED',I4,
C   *           /1X,'THE LEFT OVER POINT-SAMPLES ARE IGNORED')
C
C   GO TO 500
C
C   1 FORMAT(A1,3A4,3F8.2,2F7.4)
C   2 FORMAT(1X,3A4,2F9.3,F11.3,2F9.4,'(,2F9.3,)')
C   END

```

```

C *** SUBROUTINE PCINL ***
C
C OBJECTIVES:
C LINE DATA GROUP
C 1. READ FLIGHT LINE DATA INTO COMMON BLOCK PCLDTA
C 2. CONVERT (LAT.,LONG.) TO (X,Y) HRAP COORDINATE SYSTEMS
C
C
C SUBROUTINE PCINL(CODE,IOUNIT,INPLST)
C
C COMMON /PCLDTA/MLDTA,NLDTA,ALPHAL(100),CL(100),XLDTA(2,100),
C * * * * * YLDTA(2,100),CALDTA(100),VALL(100),DESCRL(3,100)
C
C
C DIMENSION TEMPX(500),TEMPY(500),PXY(2,500)
C
C CHARACTER CODE*1,INPLST*1
C
C DO 100 I=1,MLDTA+1
C     READ(29,1) CODE,(DESCRL(J,I),J=1,3),XLDTA(1,I),YLDTA(1,I),
C * * * * * XLDTA(2,I),YLDTA(2,I),VALL(I),CL(I),ALPHAL(I)
C     IF(CODE .NE. ' ') GO TO 150
C     IF(I .GT. 1 .AND. CL(I) .LT. .00001) CL(I)=CL(I-1)
C     IF(I .GT. 1 .AND. ALPHAL(I) .LT. .00001) ALPHAL(I)=ALPHAL(I-1)
C     NLDTA=NLDTA+1
100 CONTINUE
C     IF(NLDTA .GT. MLDTA) GO TO 9000
C
C 150 IF(INPLST .EQ. 'N') GO TO 500
C     DO 160 I=1,NLDTA
C         WRITE(IOUNIT,2) (DESCRL(J,I),J=1,3),XLDTA(1,I),YLDTA(1,I),
C * * * * * XLDTA(2,I),YLDTA(2,I),VALL(I),CL(I),ALPHAL(I)
160 CONTINUE
C
C 500 CONTINUE
C     DO 300 J=1,2
C         DO 200 I=1,NLDTA
C             TEMPX(I)=XLDTA(J,I)
C             TEMPY(I)=YLDTA(J,I)
200 CONTINUE
C
C     CALL PCHRAP(NLDTA,TEMPX,TEMPY,PXY)
C
C     DO 250 I=1,NLDTA
C         XLDTA(J,I)=PXY(1,I)
C         YLDTA(J,I)=PXY(2,I)
250 CONTINUE
C
C 300 CONTINUE
C
C     RETURN
C
C 9000 WRITE(IOUNIT,9100) MLDTA
C 9100 FORMAT(/1X,'ERROR: # OF LIN-SAMPLES CAN NOT EXCEED',I4,
C * * * * * /1X,' THE LEFT OVER LINE-SAMPLES ARE IGNORED')

```

```
C
  RETURN
C
 1 FORMAT(A1,3A4,5F8.2,2F7.4)
 2 FORMAT(1X,3A4,4F9.3,F12.4,2F9.4)
C
  END
```

```

C *** SUBROUTINE PCINA ***
C
C OBJECTIVES:
C   AREAL DATA GROUP
C     1. PROCESSES ALL AREAL SAMPLE VALUES FOR A SINGLE TECHNOLOGY TO
C        PRODUCE A BASIN-WIDE AREAL AVERAGE FOR THAT TECHNOLOGY
C     2. LOAD THE RESULTS INTO ONE ENTRY IN COMMON BLOCK PCADTA
C
C
C   SUBROUTINE PCINA(CODE, IOUNIT, INPLST, *)
C
C   COMMON /PCADTA/MADTA, NADTA, MATDTA, NATDTA, CA(20), VALA(20),
C   *          DESCRA(4,20)
C   COMMON /PCBDCH/MCHAN, NCHAN, BDCHAN(10000)
C
C   DIMENSION XADTA(4), YADTA(4)
C   DIMENSION XYGRID(2,4)
C
C   CHARACTER CODE*1, INPLST*1
C
C   100 CONTINUE
C
C   NADTA=0
C   NATDTA=NATDTA+1
C   IF(NATDTA .GT. MATDTA) GO TO 9000
C   CMSUM=0.
C   CDSUM=0.
C   CSUM=0.
C
C   READ(29,1) CODE, (DESCRA(J,NATDTA), J=1,4)
C   IF(INPLST .EQ. 'N') GO TO 300
C   WRITE(IOUNIT,3) (DESCRA(J,NATDTA), J=1,4)
300  READ(29,2) CODE, (XADTA(I), YADTA(I), I=1,4), VALA1, TEMCA1
      IF(CODE .NE. ' ') GO TO 6000
      NADTA=NADTA+1
      IF(NADTA .GT. 1 .AND. TEMCA1 .LT. .00001) TEMCA1=CA1
      CA1=TEMCA1
      IF(NADTA .GT. MADTA) GO TO 9000
C
C   CONVERT (LAT., LONG.) TO HRAP (X, Y)
C
C   CALL PCHRAP(4, XADTA, YADTA, XYGRID)
C
C   FIND SIZE OF SAMPLE AREA
C
C   CALL PCAREA(XYGRID, 4, 2, DAREA, ISW, *9000)
C   IF(ISW .EQ. -1) GO TO 9000
C
C   LOAD THE AREAL SAMPLE BOUNDARY DATA INTO ARRAY BDCHAN
C
C   LAS=IFIX(BDCHAN(2))
C
C   DO 400 J=1,4
C       BDCHAN(LAS+5+2*J)=XYGRID(1,J)
C       BDCHAN(LAS+6+2*J)=XYGRID(2,J)
400  CONTINUE

```

```

C
  IF(INPLST .EQ. 'N') GO TO 370
  WRITE(IOUNIT,4) (XADTA(I),YADTA(I),I=1,4),VALA1,CA1
  WRITE(IOUNIT,5) (BDCHAN(LAS+5+2*I),BDCHAN(LAS+6+2*I),I=1,4)
  WRITE(IOUNIT,6) DAREA
C
  370 BDCHAN(LAS+3)=4.+0.1
C
  CALL PCBDPK
C
C FIND AUGMENTED BOUNDARIES
C
C 1. ALLOCATE 1ST BOUNDARY (AUGMENTED BASIN BOUNDARY)
C
  MPTS1=(MCHAN-NCHAN-14)/6
  CALL PCBALC(2,MPTS1,LOCA1)
C
C 2. ALLOCATE 2ND BOUNDARY (AUGMENTED AREAL SAMPLE BOUNDARY)
C
  MPTS2=-1
  CALL PCBALC(2,MPTS2,LOCA2)
C
C 3. AUGMENTED BOUNDARIES
C
  NF1=IFIX(BDCHAN(4))           /* # OF VERTEX IN 1ST BOUNDARY
*/
  NF2=IFIX(BDCHAN(LAS+3))
C
  CALL PCBAUG(BDCHAN(8),NF1,BDCHAN(LAS+7),NF2,BDCHAN(LOCA1+7),
*           MPTS1,NPTS1,BDCHAN(LOCA2+7),MPTS2,NPTS2,
*           NINTER,IER,*9100)
C
  BDCHAN(LOCA1+3)=FLOAT(NPTS1)+0.1
  BDCHAN(LOCA2+3)=FLOAT(NPTS2)+0.1
C
  IF(IER .NE. 0) GO TO 9000
  IF(NINTER .GT. 0) GO TO 2000
C
C NO INTERSECTION, TEST FOR WHOLE SAMPLE INSIDE BASIN
C
  CALL PCIFIN(BDCHAN(LAS+7),BDCHAN(LAS+8),BDCHAN(8),NF1,2,INOUT)
  IF(INOUT .EQ. 0) GO TO 1200
  IF(INPLST .EQ. 'N') GO TO 1100
  WRITE(IOUNIT,1050)
1050 FORMAT(1X,'WHOLE SAMPLE INSIDE BASIN')
1100 CMSUM=VALA1*CA1+CMSUM
  CSUM=CA1+CMSUM
  CDSUM=CA1*DAREA+CDSUM
  GO TO 5000
C
C TEST FOR WHOLE BASIN INSIDE AREAL SAMPLE

```

```

C
1200 CALL PCIFIN(BDCHAN(8),BDCHAN(9),BDCHAN(LAS+7),NF2,2,INOUT)
  IF(INOUT .EQ. 0) GO TO 1270
  IF(INPLST .EQ. 'N') GO TO 1250
  WRITE(IONUNIT,1230)
1230 FORMAT(1X,'WHOLE BASIN INSIDE SAMPLE')
1250 CADJ=CA1*BDCHAN(5)/DAREA
  CMSUM=CADJ*VALA1+CMSUM
  CSUM=CADJ+CSUM
  CDSUM=CADJ*DAREA+CDSUM
  GO TO 5000
1270 IF(INPLST .EQ. 'N') GO TO 5000
  WRITE(IONUNIT,1280)
1280 FORMAT(1X,'BASIN AND AREAL SAMPLE ARE DISJOINT')
  GO TO 5000
C
2000 CALL PCBDPK
  LOCA2=IFIX(BDCHAN(LOCA1+1))
C
C ALLOCATE SPACE FOR SUB-BASIN
C
  MSUB=-1
  CALL PCBALC(1,MSUB,LOCSUB)
C
  IF(INPLST .EQ. 'N') GO TO 2090
  WRITE(IONUNIT,9)
  WRITE(IONUNIT,8) (BDCHAN(I),I=1,IFIX(BDCHAN(2))-1)
  WRITE(IONUNIT,8) (BDCHAN(I),I=IFIX(BDCHAN(2)),LOCA1-1)
  WRITE(IONUNIT,8) (BDCHAN(I),I=LOCA1,LOCA2-1)
  WRITE(IONUNIT,8) (BDCHAN(I),I=LOCA2,LOCSUB-1)
C
C
C FIND OVERLAPPING AREAS
C
2090 BID=0.
  ILOOP1=0
  ILOOP2=0
C
2100 CALL PCSUBB(BDCHAN(LOCA1+7),NPTS1,BDCHAN(LOCA2+7),NPTS2,
  *           BDCHAN(LOCSUB+7),NSUB,MSUB,1,IER,0,*9200)
  IF(NSUB .EQ. 0) GO TO 2200
  BDCHAN(LOCSUB+3)=FLOAT(NSUB)+0.1
  ILOOP1=1
  CALL PCAREA(BDCHAN(LOCSUB+7),NSUB,2,AREA,IER,*9000)
  BID=BID+AREA
  GO TO 2100
C
2200 CALL PCSUBB(BDCHAN(LOCA2+7),NPTS2,BDCHAN(LOCA1+7),NPTS1,
  *           BDCHAN(LOCSUB+7),NSUB,MSUB,1,IER,0,*9200)
  IF(NSUB .EQ. 0) GO TO 2300
  BDCHAN(LOCSUB+3)=FLOAT(NSUB)+0.1
  ILOOP2=1
  CALL PCAREA(BDCHAN(LOCSUB+7),NSUB,2,AREA,IER,*9000)
  BID=BID+AREA
  GO TO 2200

```

```

C
2300 IF(ILOOP1 .EQ. 1 .OR. ILOOP2 .EQ. 1) GO TO 2500
    CALL PCSUBB(BDCHAN(LOCA1+7),NPTS1,BDCHAN(LOCA2+7),NPTS2,
    *          BDCHAN(LOCSUB+7),NSUB,MSUB,1,IER,1,*9200)
    IF(NSUB .EQ. 0) GO TO 2350
    BDCHAN(LOCSUB+3)=FLOAT(NSUB)+0.1
    CALL PCAREA(BDCHAN(LOCSUB+7),NSUB,2,AREA,IER,*9000)
    BID=BID+AREA
    GO TO 2500
2350 WRITE(IOUNIT,2360)
2360 FORMAT(/1X,'* INTERSECTION PTS FOUND, BUT SUB-BASIN NOT FOUND *')
    GO TO 5000
C
C BID IS OVERLAPPED AREA, UPDATE SUMS
C
2500 CADJ=CA1*BID/DAREA
    CMSUM=CADJ*VALA1+CMSUM
    CSUM=CADJ+CSUM
    CDSUM=CADJ*DAREA+CDSUM
    GO TO 5000
C
C DEFINE VALUE FOR THIS TECHNOLOGY
C
6000 IF(CSUM .EQ. 0) GO TO 6100
    VALA(NATDTA)=CMSUM/CSUM
    CA(NATDTA)=CDSUM/BDCHAN(5)
    GO TO 5000
C
6100 VALA(NATDTA)=0.
    CA(NATDTA)=CDSUM/BDCHAN(5)
    GO TO 5000
C
C CLEANUP: JUST DELETES SPACE ALLOCATED FOR AUGMENTED BOUNDARIES
C
5000 IF(LOCA1 .GT. 0) BDCHAN(LOCA1)==-1.0
    IF(LOCA2 .GT. 0) BDCHAN(LOCA2)==-1.0
    IF(LOCSUB .GT. 0) BDCHAN(LOCSUB)==-1.0
C
    CALL PCBDPK
C
C
    IF(CODE .EQ. ' ') GO TO 300
    IF(CODE .EQ. 'A') GO TO 100
    RETURN
C
C ERROR
C
9000 WRITE(IOUNIT,9010)
9010 FORMAT(/1X,'THERE IS A PROBLEM CALLING PCAREA IN AREAL SAMPLES')
    RETURN1
C
9100 WRITE(IOUNIT,9110)
9110 FORMAT(/1X,'THERE IS A PROBLEM IN CALLING PCBAUG')
    RETURN1

```

```
C
9200 WRITE( IOUNIT,9210)
9210 FORMAT(/1X,'THERE IS A PROBLEM IN CALLING PCSUBB')
      RETURN1
C
1 FORMAT(A1,4A4)
2 FORMAT(A1,9F8.2,F7.4)
3 FORMAT(/1X,4A4)
4 FORMAT(/1X,9F8.2,F7.4)
5 FORMAT('(',8F8.2,')')
6 FORMAT(1X,'AREA = ',F14.4)
8 FORMAT(1X,8F9.2)
9 FORMAT(1X,'AUGMENTED BOUNDARIES:')
C
      END
```

```

C *** SUBROUTINE PCINI ***
C
C OBJECTIVES:
C   IMAGE DATA GROUP
C     1. PROCESSES ALL IMAGERY SAMPLE VALUES FOR A SINGLE TECHNOLOGY TO
C        PRODUCE AN IMAGERY AVERAGE FOR THAT TECHNOLOGY
C     2. LOADS THE RESULTS INTO ONE ENTRY IN COMMON BLOCK PCADTA
C
C
C   SUBROUTINE PCINI(CODE, IOUNIT, INPLST, *)
C
C   COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
C   *           DESCRA(4,20)
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
C   DIMENSION PV(64)
C   CHARACTER*1 PVCC(64),CHARC(64),INPLST*1
C   DIMENSION XYGRID(2,1)
C
C   CHARACTER CODE*1
C
C 100 CONTINUE
C   NADTA=0
C   NATDTA=NATDTA+1
C   IF(NATDTA .GT. MATDTA) GO TO 9000
C
C   READ(29,1) CODE,(DESCRA(J,NATDTA),J=1,4),C
C   READ(29,3) CODE,NOVAL,(PVCC(I),PV(I),I=1,NOVAL)
C
C   IF(INPLST .EQ. 'N') GO TO 200
C   WRITE(IOUNIT,2) (DESCRA(J,NATDTA),J=1,4),C
C   WRITE(IOUNIT,4) NOVAL,(PVCC(I),PV(I),I=1,NOVAL)
C
C 200 SUMPV=0.
C   NPIB=0           /* # OF PIXELS IN BASIN */
C
C 300 READ(29,6) CODE,XST,YST,XEND,YEND,NPL,(CHARC(I),I=1,NPL)
C
C   IF(CODE .NE. ' ') GO TO 6000
C   NADTA=NADTA+1
C   IF(NADTA .GT. MADTA) GO TO 9000
C
C   IF(INPLST .EQ. 'N') GO TO 400
C   WRITE(IOUNIT,7) XST,YST,XEND,YEND,NPL,(CHARC(I),I=1,NPL)
C
C   CONVERT (LAT.,LONG.) TO HRAP (X,Y)
C
C 400 CALL PCHRAP(1,XST,YST,XYGRID)
C   XST=XYGRID(1,1)
C   YST=XYGRID(2,1)
C
C   CALL PCHRAP(1,XEND,YEND,XYGRID)
C   XEND=XYGRID(1,1)
C   YEND=XYGRID(2,1)

```

```

C
  IF(INPLST .EQ. 'N') GO TO 450
  WRITE(IONUNIT,8) XST,YST,XEND,YEND
C
  450 DX=(XEND-XST)/(NPL-1)
      DY=(YEND-YST)/(NPL-1)
C
C PROCESS A LINE OF IMAGE
C
  DO 500 I=1,NPL
      XI=XST+DX*(I-1)
      YI=YST+DY*(I-1)
C
  DO 550 J=1,NOVAL
      IF(CHARC(I) .EQ. PVCC(J)) GO TO 560
  550 CONTINUE
      GO TO 9000
  560 PVI=PV(J)
      NF1=IFIX(BDCHAN(4))
      CALL PCIFIN(XI,YI,BDCHAN(8),NF1,2,INOUT)
      IF(INOUT .LT. 1) GO TO 500
      SUMPV=SUMPV+PVI
      NPIB=NPIB+1
  500 CONTINUE
C
      GO TO 300
C
C DEFINE VALUE FOR THIS TECHNOLOGY
C
  6000 IF(NPIB .EQ. 0) GO TO 6100
      VALA(NATDTA)=SUMPV/NPIB
      CA(NATDTA)=C
C
      IF(CODE .EQ. 'I') GO TO 100          /* ANOTHER TECHNOLOGY */
      RETURN
C
  6100 VALA(NATDTA)=0.
      CA(NATDTA)=0.
      IF(CODE .EQ. 'I') GO TO 100
      RETURN
C
      1 FORMAT(A1,4A4,F7.4)
      2 FORMAT(/1X,4A4,F7.4)
      3 FORMAT(A1,I4,8(A1,F8.2),7(/5X,8(A1,F8.2)))
      4 FORMAT(1X,I4,8(A1,F8.2),7(/5X,8(A1,F8.2)))
      6 FORMAT(A1,4F8.2,I4,43A1)
      7 FORMAT(1X,4F8.2,I4,43A1)
      8 FORMAT('(',4F8.2,')')
  9000 WRITE(IONUNIT,9100)
  9100 FORMAT(/1X,'THERE IS A PROBLEM IN IMAGERY SAMPLES')
C
      RETURN1
      END

```

```

C *** SUBROUTINE PCSUBB ***
C
C OBJECTIVES:
C
C PROCESSES TWO AUGMENTED BOUNDARY SETS (FROM SUBROUTINE PCBAUG) TO
C FIND ONE SUB-BASIN
C TO FIND ALL SUB-BASINS INSIDE BOTH BOUNDARIES, PCSUBB MUST BE
C CALLED REPEATEDLY UNTIL NSUBB=0
C
C PRINCIPAL VARIABLES:
C
C   NAME  I/O  DIMENSION  DESCRIPTION
C   ----  ---  -----  -----
C   G1    I    (3,NG1)  AUGMENTED BOUNDARY 1
C   NG1   I    -        # OF VERTICES IN G1
C   G2    I    (3,NG2)  AUGMENTED BOUNDARY 2
C   NG2   I    -        # OF VERTICES IN G2
C   SUBB  O    (2,MSUBB) VERTEX PTS OF A SUB-BASIN INSIDE BOTH BOUNDARIES
C                           1 AND 2 (FOR INOUT=1)
C                           FOR INOUT=0, SUBB DEFINES A SUB-BASIN INSIDE
C                           BOUNDARY1 BUT NOT INSIDE BOUNDARY 2
C   NSUBB O    -        # OF VERTICES IN SUBB
C   MSUBB I    -        MAX # OF VERTICES IN SUBB
C   INOUT I    -        =1 (NORMAL) FIND A SUB-BASIN INSIDE BOTH 1 & 2
C                           =0 TO FIND A SUB-BASIN INSIDE 1 BUT OUTSIDE 2
C   IER   O    -        ERROR FLAG, =0 NORMAL RETURN
C                           =1 ERROR - NOT ENOUGH SPACE
C   IVPIB I    -        =0 SOME VERTEX-PTS INSIDE BOUNDARY OF OTHER
C                           =1 NO VERTEX-PT INSIDE BOUNDARY OF EACH OTHER
C
C
C SUBROUTINES CALLED:
C   PCIFIN
C
C
C   SUBROUTINE PCSUBB(G1,NG1,G2,NG2,SUBB,NSUBB,MSUBB,INOUT,IER,
C   *           IVPIB,*)
C
C   DIMENSION G1(3,NG1),G2(3,NG2),SUBB(2,MSUBB)
C
C   IER=0
C   NSUBB=0
C
C FIND A STARTING POINT
C
C   IF(IVPIB .EQ. 1) GO TO 511
C   DO 500 IST=1,NG1
C     IF(ABS(G1(3,IST)) .GT. 0.1) GO TO 500          /* INTERESCTED OR USED
*/
C
C   CALL PCIFIN(G1(1,IST),G1(2,IST),G2,NG2,3,ITST)
C   IF(ITST .EQ. INOUT) GO TO 550
C   500 CONTINUE
C
C   RETURN

```

```

C
511 DO 521 IST=1,NG1
    IEND=IST+1
    IF(IEND .GT. NG1) IEND=1
    IF(G1(3,IST) .GT. 0.1 .AND. G1(3,IEND) .GT. 0.1) GO TO 531
521 CONTINUE
C
    RETURN
C
531 ISTG2=IFIX(G1(3,IST))
C
C START POINT FOUND
C
550 I=IST
    ISET=1
C
C ADD POINT I
C
600 IF(NSUBB .EQ. MSUBB) GO TO 9000
    NSUBB=NSUBB+1
    DO 1000 J=1,2
        IF(ISET .EQ. 1) SUBB(J,NSUBB)=G1(J,I)
        IF(ISET .EQ. 2) SUBB(J,NSUBB)=G2(J,I)
1000 CONTINUE
C
C NOW MARK THIS POINT AS USED
C
    IF(ISET .NE. 1) GO TO 1050
    IF(G1(3,I) .LT. 0.1) G1(3,I)=-1.0
    GO TO 1100
1050 IF(ISET .NE. 2) GO TO 1150
    IF(G2(3,I) .LT. 0.1) G2(3,I)=-1.0
C
    1100 IF(ISET .EQ. 2) GO TO 1200
C
C FOR SET 1
C
1150 I=I+1
    IF(I .GT. NG1) I=1
    IF(I .EQ. IST) RETURN
C
C CHECK FOR INTERSECTION
C
    IF(G1(3,I) .LT. 0.1) GO TO 600
    ISET=2
    I=IFIX(G1(3,I))
    GO TO 600
C
C FOR SET 2
C
1200 I=I+1
    IF(INOUT .EQ. 0) I=I-2
    IF(I .LT. 1) I=NG2
    IF(I .GT. NG2) I=1
    IF(IVPIB .EQ. 1 .AND. I .EQ. ISTG2) RETURN

```

```
C
C CHECK FOR INTERSECTION
C
IF(G2(3,I) .LT. 0.1) GO TO 600
ISET=1
I=IFIX(G2(3,I))
GO TO 600
C
C
C ERPOR: NOT ENOUGH SPACE FOR SUB-BASIN POINT
C
9000 IER=-1
RETURN1
END
```

```

C *** PCBAUG ***
C
C OBJECTIVES:
C   PRODUCES THE AUGMENTED BOUNDARY SETS WHICH INCLUDE ALL INTERSECTION
C   POINTS OF TWO BOUNDARIES
C
C PRINCIPAL VARIABLES:
C   NAME  I/O  DIMENSION  DESCRIPTION
C   ----  ---  -----  -----
C   F1XY   I   (2,NF1)   X-Y PAIRS OF FIRST BOUNDARY (1=X, 2=Y)
C   NF1    I   -          # OF PAIRS IN FIRST BOUNDARY
C   F2XY   I   (2,NF2)   X-Y PAIRS OF SECOND BOUNDARY
C   NF2    I   -          # OF PAIRS IN SECOND BOUNDARY
C   G1XY   O   (3,MG1)   AUGMENTED FIRST BOUNDARY INCLUDING INTERSECTIONS
C                      1=X,2=Y,3=FLAY
C   MG1    I   -          MAX. # OF POINTS IN AUGMENTED FIRST BOUNDARY
C   NG1    O   -          ACTUAL # OF PTS IN COMPLETED FIRST BOUNDARY
C   G2XY   O   (3,MG2)   SECOND AUGMENTED BOUNDARY
C   MG2    I   -          MAX. # OF POINTS
C   NG2    O   -          ACTUAL # OF PTS
C   NINTER O   -          # OF INTERSECTIONS BETWEEN F1XY AND F2XY
C   IER    O   -          RETURN STATUS, =0 NO ERROR
C                      =1 ERROR OCCURRED (TOO LITTLE
C   STORAGE)
C
C
C   SUBROUTINE PCBAUG(F1XY,NF1,F2XY,NF2,G1XY,MG1,NG1,G2XY,MG2,NG2,
C   *                  NINTER,IER,*)
C
C
C   DIMENSION F1XY(2,NF1),F2XY(2,NF2),G1XY(3,MG1),G2XY(3,MG2)
C   DIMENSION G2XY0(1000)
C
C
C   IER=0
C   NINTER=0
C   DO 200 I=1,NF1
C       DO 100 J=1,2
C 100   G1XY(J,I)=F1XY(J,I)
C 200   G1XY(3,I)=0.
C       DO 250 I=1,NF2
C           DO 150 J=1,2
C 150   G2XY(J,I)=F2XY(J,I)
C 250   G2XY(3,I)=0.
C
C   NG1=NF1
C   NG2=NF2
C
C FIND ALL INTERSECTIONS, ADD TO G1XY & G2XY

```

```

C
1000 CONTINUE
  DO 1900 IS=1,NG1
    IE=IS+1
    IF(IE .GT. NG1) IE=1
    /* ENDING POINT OF A LINE */
    X1S=G1XY(1,IS)
    X1E=G1XY(1,IE)
    IFL1S=IFIX(G1XY(3,IS))
    Y1S=G1XY(2,IS)
    Y1E=G1XY(2,IE)
    IFL1E=IFIX(G1XY(3,IE))
    A1=X1S-X1E
    B1=Y1S-Y1E
    C1=X1S*Y1E-Y1S*X1E

C
  DO 1800 JS=1,NG2
    JE=JS+1
    IF(JE .GT. NG2) JE=1
    IF(IFL1S .EQ. 0 .AND. IFL1E .EQ. 0) GO TO 1300
    IFL2S=IFIX(G2XY(3,JS))
    IFL2E=IFIX(G2XY(3,JE))
    IF(IFL1S .EQ. 0) GO TO 1200
    IF(IFL1S .EQ. IFL2S .OR. IFL1S .EQ. IFL2E) GO TO 1800
1200    IF(IFL1E .EQ. 0) GO TO 1300
    IF(IFL1E .EQ. IFL2S .OR. IFL1E .EQ. IFL2E) GO TO 1800
C
1300    X2S=G2XY(1,JS)
    X2E=G2XY(1,JE)
    Y2S=G2XY(2,JS)
    Y2E=G2XY(2,JE)
    A2=X2S-X2E
    B2=Y2S-Y2E
    C2=X2S*Y2E-Y2S*X2E
    D=B2*A1-B1*A2

C
    IF(D .EQ. 0.0) GO TO 1800
    XSTAR=(C1*A2-C2*A1)/D
    IF((XSTAR-X1S)*(XSTAR-X1E) .GT. 0.0) GO TO 1800
    IF((XSTAR-X2S)*(XSTAR-X2E) .GT. 0.0) GO TO 1800
    YSTAR=(C1*B2-C2*B1)/D
    IF((YSTAR-Y1S)*(YSTAR-Y1E) .GT. 0.0) GO TO 1800
    IF((YSTAR-Y2S)*(YSTAR-Y2E) .GT. 0.0) GO TO 1800

C
C ADD POINT (XSTAR,YSTAR) TO G1XY,G2XY AND RE-ORDERING
C
    IF(NG1 .GE. MG1 .OR. NG2 .GE. MG2) GO TO 1700
C
    DO 1500 I=NG1,IS,-1
      DO 1500 J=1,3
        G1XY(J,I+1)=G1XY(J,I)
1500    CONTINUE
C
    DO 1600 I=NG2,JS,-1
      DO 1600 J=1,3
        G2XY(J,I+1)=G2XY(J,I)
1600    CONTINUE

```

```

C
NINTER=NINTER+1
FLAG=FLOAT(NINTER)+0.1
G1XY(1,IS+1)=XSTAR
G1XY(2,IS+1)=YSTAR
G1XY(3,IS+1)=FLAG
G2XY(1,JS+1)=XSTAR
G2XY(2,JS+1)=YSTAR
G2XY(3,JS+1)=FLAG
NG1=NG1+1
NG2=NG2+1
GO TO 1000
C
1700 IER=1
RETURN1
C
1800 CONTINUE
1900 CONTINUE
C
C RENUMBER FLAGS OF ALL INTERSECTION POINTS
C
IF(NINTER .EQ. 0) RETURN
C
DO 2000 K=1,NG2
2000 G2XY0(K)=G2XY(3,K)
C
DO 2500 I=1,NG1
IFL1=IFIX(G1XY(3,I))
IF(IFL1 .EQ. 0) GO TO 2500
DO 2400 J=1,NG2
IFL2=IFIX(G2XY0(J))
IF(IFL2 .NE. IFL1) GO TO 2400
G1XY(3,I)=FLOAT(J)+0.1
G2XY(3,J)=FLOAT(I)+0.1
C
GO TO 2500
2400 CONTINUE
C
C
C
C IF FULL THROUGH J LOOP: ERROR, DID NOT FIND INTERSECTION PT IN 2ND LIST
C
C     WRITE(1,1)
C     1 FORMAT(/1X,'ERROR: DID NOT FIND INTERSECTION PT IN 2ND LIST')
RETURN1
C
2500 CONTINUE
C
C
RETURN
END

```

```

C *** SUBROUTINE PCINE ***
C
C OBJECTIVES:
C   END OF DATA CASE
C     1. COMBINES ALL AREAL AND IMAGE TECHNOLOGIES IN COMMON BLOCK PCADTA
C        INTO A SINGLE BASIN ESTIMATE (BY CALLING PCCOMB)
C     2. USES GRID METHOD TO COMBINE ALL SAMPLING TECHNOLOGIES (POINT, LINE
C        AND AREAL) (BY CALLING PCGRID)
C     3. OUTPUTS RESULTS
C     4. SETS CODE INDICATING THAT DATA INPUT IS NOT VALID UNTIL AFTER A
C        CONTROL CODE 'N'
C
C
C SUBROUTINE PCINE(CODE,TITLE,IOUNIT,*)
C
C   COMMON /PCPDTA/MPDTA,NPDTA,ALPHAP(500),CP(500),XPDTA(500),
C   *           YPDTA(500),CAPDTA(500),VALP(500),DESCRP(3,500)
C   COMMON /PCLDTA/MLDTA,NLDTA,ALPHAL(100),CL(100),XLDTA(2,100),
C   *           YLDTA(2,100),CALDTA(100),VALL(100),DESCRL(3,100)
C   COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
C   *           DESCRA(4,20)
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
C   CHARACTER CODE*1,TITLE*79
C
C COMBINE ALL EXISTING AREAL AND IMAGE TECHNOLOGIES IN COMMON BLOCK
C PCADTA INTO A SINGLE BASIN ESTIMATE
C
C   CALL PCCOMB(IOUNIT)
C
C   WRITE(1,10)
10  FORMAT(/1X,'ENTER # OF GRID POINTS (INTEGER IN FOUR DIGITS)')
11  READ(1,12) NGRID
12  FORMAT(I4)
C
C   IF(NGRID .LE. 0) GO TO 1000
C   CALL PCGRID(NGRID,AVAL,ACORR,TITLE,IOUNIT,*9900)
C
C   WRITE(IOUNIT,5) AVAL,ACORR
5   FORMAT(//1X,'THE BASIN-WIDE AREAL AVERAGE ESTIMATE = ',F12.4,
*           /1X,'THE OVERALL ACCURACY OF THIS ESTIMATE = ',F6.4)
C
C   WRITE(1,14)
14  FORMAT(//1X,'ENTER ANOTHER # OF GRID POINTS (IF NO MORE,
*           ' ENTER 0)')
      GO TO 11
C
1000 READ(29,1) CODE
1  FORMAT(A1)
      IF(CODE .EQ. 'C' .OR. CODE .EQ. 'N' .OR. CODE .EQ. 'S') RETURN
      WRITE(IOUNIT,7)
7   FORMAT(//1X,'DATA CASE INPUT IS NOT VALID')
9900 CONTINUE
      RETURN1
C
C   END

```

```

C ** SUBROUTINE PCCOMB ***
C
C OBJECTIVES:
C   COMBINES ALL AVAILABLE AREAL AND IMAGERY SAMPLING TECHNOLOGIES
C   INTO A SINGLE BASIN-WIDE AREAL ESTIMATE
C
C
C   SUBROUTINE PCCOMB(IOUNIT)
C
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C   COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
C   *           DESCRA(4,20)
C
C INITIALIZE
C
C   BDCHAN(6)=-1.0
C   BDCHAN(7)=0.0
C
C   IF(NATDTA .LT. 1) RETURN
C   WRITE(IOUNIT,1)
C   SUM1=0.0
C   SUMBM=0.0
C   SUM2=0.0
C
C FIND A FACTOR
C
C   DO 500 I=1,NATDTA
C     IF(CA(I) .GT. 0.999) CA(I)=.999
C     IF(CA(I) .LT. 0.001) CA(I)=.001
C     SUM1=SUM1+(CA(I)/(1-CA(I)))
C   500 CONTINUE
C
C WEIGHT EACH MEASUREMENT
C
C   DO 600 I=1,NATDTA
C     BI=CA(I)/(SUM1*(1.-CA(I)))
C     SUMBM=BI*VALA(I)+SUMBM
C     SUM2=BI/SUM1+SUM2
C     WRITE(IOUNIT,3) I,CA(I),VALA(I),BI
C   600 CONTINUE
C
C COMBINE
C
C   BDCHAN(7)=SUMBM
C   BDCHAN(6)=1.0/(1.+SUM2)
C
C   1 FORMAT(/1X,'AREAL SAMPLING SUMMARY:',
C   *          /3X,'SAMPLING TECH   CORRELATION      VALUE      WEIGHT',
C   *          /3X,'-----  -----  -----  -----')
C   3 FORMAT(/8X,I2,11X,F7.4,5X,F8.2,4X,F7.4)
C   RETURN
C   END

```

```

C     ***   SUBROUTINE  PCSYMBO   ***
C
C
C     SUBROUTINE PCSYMBO(NX,IHTYP,IHNUM,IOUNIT)
C
C     DIMENSION IHTYP(NX),IHNUM(NX)
C     CHARACTER*1 TYPE(130),LCHAR(130)*1,NCHAR(130)*1
C
C     DATA LCHAR/'A','B','C','D','E','F','G','H','I','J','K','L','M',
C     *           'N','O','P','Q','R','S','T','U','V','W','X','Y',105*'Z'/
C     *           DATA NCHAR/'1','2','3','4','5','6','7','8','9','0','#','&','
C     *           '+','-', '=','?','@','!','/','111*%'/
C
C     DO 9000 I=1,NX
C     IF(IHTYP(I) .NE. 0) GO TO 5000
C     TYPE(I)='*'
C     GO TO 9000
C 5000 CONTINUE
C     GO TO (1000,2000,3000), IHTYP(I)
C
C 1000  TYPE(I)='$'
C     GO TO 9000
C
C 2000  TYPE(I)=LCHAR(IHNUM(I))
C     GO TO 9000
C
C 3000  TYPE(I)=NCHAR(IHNUM(I))
C
C 9000 CONTINUE
C
C     WRITE(IOUNIT,9900) (TYPE(K),K=1,NX)
C 9900 FORMAT(1X,130A1)
C     RETURN
C     END

```

```

C *** SUBROUTINE PCGRID ***
C
C OBJECTIVES:
C   USING GRID METHOD, COMBINES ALL DATA (POINT, LINE, AREAL, IMAGERY)
C   INTO A SINGLE BASIN-WIDE ESTIMATE
C
C
C   SUBROUTINE PCGRID(NGRID,AVAL,ACORR,TITLE,IOUNIT,*)
C
C   COMMON /PCPDTA/MPDTA,NPDTA,ALPHAP(500),CP(500),XPDTA(500),
C   *           YPDTA(500),CAPDTA(500),VALP(500),DESCRP(3,500)
C   COMMON /PCLDTA/MLDTA,NLDTA,ALPHAL(100),CL(100),XLDTA(2,100),
C   *           YLDTA(2,100),CALDTA(100),VALL(100),DESCRL(3,100)
C   COMMON /PCADTA/MADTA,NADTA,MATDTA,NATDTA,CA(20),VALA(20),
C   *           DESCRA(4,20)
C   COMMON /PCBDCH/MCHAN,NCHAN,BDCHAN(10000)
C
C   CHARACTER ANSWER*1,TITLE*79,LCHAR(130)*1,NCHAR(130)*1
C
C   DIMENSION IHYP(130),IHN(130)
C
C   DATA LCHAR/'A','B','C','D','E','F','G','H','I','J','K','L','M',
C   *           'N','O','P','Q','R','S','T','U','V','W','X','Y',105*'Z'/
C   DATA NCHAR/'1','2','3','4','5','6','7','8','9','0','#','&','
C   *           '+','-', '=','?','@','!','/','111*'%'/
C
C   C VALIDITY TESTS
C
C   IF(NGRID .LT. 100) GO TO 9000
80 IF(BDCHAN(4) .LT. 3) GO TO 9100
     IF(NPDTA+NLDTA+NATDTA .LT. 1) GO TO 9200
C
C   C DEFINE GRID
C   FIND THE WESTMOST, EASTMOST, SOUTHERN, AND NORTHERN POINTS OF THE
C   BASIN BOUNDARY RESPECTIVELY
C
C   XMIN=BDCHAN(8)
C   XMAX=BDCHAN(8)
C   YMIN=BDCHAN(9)
C   YMAX=BDCHAN(9)
C   DO 100 I=1,BDCHAN(4)
C     XMIN=AMIN1(XMIN,BDCHAN(6+2*I))
C     XMAX=AMAX1(XMAX,BDCHAN(6+2*I))
C     YMIN=AMIN1(YMIN,BDCHAN(7+2*I))
C     YMAX=AMAX1(YMAX,BDCHAN(7+2*I))
100 CONTINUE
C
C   XLEN=XMAX-XMIN
C   YLEN=YMAX-YMIN
C   DELTA=SQRT(XLEN*YLEN/NGRID)
C   NX=IFIX(XLEN/DELTA+0.5)
C   NY=IFIX(YLEN/DELTA+0.5)
C   NPB=BDCHAN(4)
C
C   C INITIALIZE

```

```

C
      NGIB=0                      /* # OF GRIDS IN BASIN */
      CSUM=0.0                     /* SUM OF GRIDS WTS */
      DO 600 I=1,NLDTA
      600  CALDTA(I)=0.0           /* FOR ALL LINE SAMPLES */
      DO 700 I=1,NPDTA
      700  CAPDTA(I)=0.0           /* FOR ALL POINT SAMPLES */
      CAASAM=0.0                   /* SUM OF G.W. FOR AREAL TECH. */
C
C GRID POINT LOOP
C
      WRITE(IONUNIT,800)
      800 FORMAT(//1X,'SAMPLING TECH DISTRIBUTION IN THE BASIN',/)
C
      DO 2900 IY=1,NY
          YGRID=YMAX-0.5*DELTA-DELTA*(IY-1)
          DO 2800 IX=1,NX
              XGRID=XMIN+0.5*DELTA+DELTA*(IX-1)
C
C TEST FOR INSIDE BASIN
C
      IHTYP(IX)=0
      CALL PCIFIN(XGRID,YGRID,BDCHAN(8),NPB,2,INOUT)
      IF(INOUT .EQ. 0) GO TO 2800
C
C AREAL DATA TESTING
C
      CHIGH=BDCHAN(6)
      IHTYP(IX)=1
      IHNUM(IX)=0
C
C POINT DATA (CHECK FOR THE MOST HIGHLY CORRELATED SAMPLE)
C
      DO 2100 I=1,NPDTA
          XOF=XGRID-XPDTA(I)
          YOF=YGRID-YPDTA(I)
          D=SQRT(XOF*XOF+YOF*YOF)
          C=CP(I)*EXP(-ALPHAP(I)*D)
          IF(C .LT. CHIGH) GO TO 2100
          CHIGH=C
          IHTYP(IX)=2
          IHNUM(IX)=I
      2100    CONTINUE
C
C LINE DATA (CHECK FOR THE MOST HIGHLY CORRELATED SAMPLE)
C
      DO 2250 I=1,NLDTA
          IF(CL(I) .LT. CHIGH) GO TO 2250
          X1=XLDTA(1,I)
          Y1=YLDTA(1,I)
          X2=XLDTA(2,I)
          Y2=YLDTA(2,I)
          R1SQ=(XGRID-X1)**2+(YGRID-Y1)**2
          R2SQ=(XGRID-X2)**2+(YGRID-Y2)**2
          R3SQ=(X1-X2)**2+(Y1-Y2)**2
          RSQMX=AMAX1(R1SQ,R2SQ)

```

```

RSQMN=AMIN1(R1SQ,R2SQ)
IF(RSQMX .LT. (RSQMN+R3SQ)) GO TO 2210 /* CLOSER TO LINE */
D=SQRT(RSQMN) /* CLOSER TO END PT */
GO TO 2220
2210 R1=SQRT(R1SQ)
R2=SQRT(R2SQ)
R3=SQRT(R3SQ)
S=(R1+R2+R3)/2
D=2.*SQRT(S*(S-R1)*(S-R2)*(S-R3))/R3
2220 C=CL(I)*EXP(-ALPHAL(I)*D) /* DISTANCE FOUND */
IF(C .LT. CHIGH) GO TO 2250
CHIGH=C
IHTYP(IX)=3
IHNUM(IX)=I
2250 CONTINUE
C
C MOST HIGHLY CORRELATED SAMPLE NOW FOUND
C
NGIB=NGIB+1
CSUM=CSUM+CHIGH
C
GO TO (2301,2302,2303),IHTYP(IX)
C
2301 CAASAM=CAASAM+CHIGH
GO TO 2800
C
2302 CAPDTA(IHNUM(IX))=CAPDTA(IHNUM(IX))+CHIGH
GO TO 2800
C
2303 CALDTA(IHNUM(IX))=CALDTA(IHNUM(IX))+CHIGH
C
C THIS GRID POINT FINISHED
C
2800 CONTINUE
C
CALL PCSYMB0(NX,IHTYP,IHNUM,IOUNIT)
C
WRITE(IOUNIT,2950) (IHTYP(K),K=1,NX)
C
2900 CONTINUE
C
C2950 FORMAT(1X,130A1)
C
C ALL GRID POINTS FINISHED
C
WRITE(IOUNIT,5556) NGIB
WRITE(IOUNIT,5558) DELTA
5558 FORMAT(1X,'THE WIDTH OF THE GRID = ',F12.4)
5556 FORMAT(/1X,'# OF GRID POINTS ON THE BASIN = ',I5)
C
ACORR=CSUM/NGIB
C
C FOR THOSE GRID PTS THAT ASSOCIATED WITH AREAL SAMPLE
C
AVAL=BDCHAN(7)*CAASAM/CSUM

```

```

C
      WRITE(IONUNIT,11) TITLE
11  FORMAT(//1X,A79,
*           //1X,2X,'SAMPLING TECH',3X,'MEASUREMENT VALUE',3X,'WEIGHT',
*           3X,'CORRELATION',3X,'ALPHA',3X,'SYMBOL',
*           /1X,2X,'-----',3X,'-----',3X,'-----',
*           3X,'-----',3X,'-----',3X,'-----',/)

C
      WRITE(IONUNIT,15) BDCHAN(7),CAASAM/CSUM,BDCHAN(6)
15  FORMAT(1X,6X,'A',13X,F12.4,6X,F6.4,5X,F6.3,5X,6X,4X,'($)',/)
16  FORMAT(1X,6X,'P',13X,F12.4,6X,F6.4,5X,F6.3,6X,F5.3,4X,'(',A1,')')
17  FORMAT(1X,6X,'L',13X,F12.4,6X,F6.4,5X,F6.3,6X,F5.3,4X,'(',A1,')')

C
C FOR THOSE GRID PTS THAT ASSOCIATED WITH POINT SAMPLE
C
      DO 3000 I=1,NPDTA
          AVAL=AVAL+VALP(I)*CAPDTA(I)/CSUM
          WRITE(IONUNIT,16)VALP(I),CAPDTA(I)/CSUM,CP(I),ALPHAP(I),
*                           LCHAR(I)
3000 CONTINUE
      WRITE(IONUNIT,3010)
3010 FORMAT(1X,' ')
C
C FOR THOSE GRID PTS THAT ASSOCIATED WITH LINE SAMPLE
C
      DO 4000 I=1,NLDTA
          AVAL=AVAL+VALL(I)*CALDTA(I)/CSUM
          WRITE(IONUNIT,17)VALL(I),CALDTA(I)/CSUM,CL(I),ALPHAL(I),
*                           NCHAR(I)
          GO TO 4000
4000 CONTINUE
C
      RETURN
C
      9000 WRITE(1,9010) NGRID
9010 FORMAT(/1X,'NO. OF GRID POINTS IS ',I4,' NOT LARGE ENOUGH',
*           /1X,'DO YOU WANT TO CONTINUE? ENTER '' '' IF YES')
      READ(1,9015) ANSWER
9015 FORMAT(A1)
      IF(ANSWER .EQ. ' ') GO TO 80
      RETURN1
C
      9100 WRITE(IONUNIT,9110)
9110 FORMAT(/1X,'BASIN BOUNDARY POINTS NOT WELL DEFINED')
      RETURN1
C
      9200 WRITE(IONUNIT,9210)
9210 FORMAT(/1X,'NO DATA AVAILABLE')
      RETURN1
C
      END

```

```
C *** SUBROUTINE PCINS ***
C
C OBJECTIVES:
C   STOP PROGRAM - NO OTHER INPUT
C
C
C   SUBROUTINE PCINS(IOUNIT)
C
C
C   WRITE(IOUNIT,1)
1 FORMAT(//15X,'*** END ***')
C
C   RETURN
C   END
```

APPENDIX B  
ERRATA  
FOR  
COMBINING REMOTELY SENSED AND OTHER  
MEASUREMENTS FOR HYDROLOGIC AREAL AVERAGES  
NASA-CR170457 OCTOBER 31, 1982

Page No.

- |    |   |  |
|----|---|--|
| 16 | 1st line  | Paramater rather than parameters   |
| 29 | 3rd para<br>2nd sent                            | should read --Advanced very high resolution radiometers (AVHRR) ...                        |
| 30 | 1st para<br>last sent                           | ADD -- in the visible and 8 km in the thermal infrared.                                    |
| 38 | sect. 5.1                                       | Change second bullet to read:<br>• NOAA-8 AVHRR, and                                       |
| 39 | 2nd para<br>1st sent                            | Change to read: The NOAA-8AVHRR on the TIROS satellites...                                 |
| 42 | Eq(3.1)   | Change last part of right hand to read: $\left[ \frac{(1-A_{\max}-A_{\min})}{A_L} \right]$ |
| 60 | Eq(3.5)   | Eliminate second = sign  |
| 64 | 2nd para<br>2nd line                            | put a subscript $\ell$ with C at end of sentence   |
| 73 | 3rd para<br>3rd sent                            | should read: The 17 points along the line...   |
| 77 | last para<br>2nd sent                           | delete the phase -- can be increased.  |
| 78 | 2nd para<br>4th line<br>5th, 7th &<br>9th lines | Schmugge not Schmugg<br>T's should have subscript B  |
| 81 | 2d & 3d para                                    | The two $\alpha$ 's should be $\sigma$ 's  |
| 82 | 2d para<br>3rd line                             | spelling of function   |

- B-1 1st para  
5th sent spelling of approach
- C-1 last sent means rather than mean
- D-19 Eq (D.33) should read:  
$$s_i = \frac{R^2}{2} (\theta_2 - \theta_1)$$
- F-17 Eq (F.19) 1st  $\beta$  below line should be  
squared ( $\beta^2$ )

# BIBLIOGRAPHIC DATA SHEET

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle  A Method to Combine Remotely Sensed and Insitu Measurements: Program Documentation		5. Report Date May 1984	6. Performing Organization Code
7. Author(s) E. L. Peck, E. R. Johnson and M. Y. Wong		8. Performing Organization Report No.	
9. Performing Organization Name and Address  Hydex Corporation 11150 Main Street, Suite 502 Fairfax, Virginia 22030		10. Work Unit No.	11. Contract or Grant No. NAS5-27554
12. Sponsoring Agency Name and Address  Dr. Thomas Schmugge National Aeronautics and Space Administration Goddard Space Flight Center Greenbelt Road, Greenbelt, Maryland 20771		13. Type of Report and Period Covered  Interim Report	14. Sponsoring Agency Code
15. Supplementary Notes  Support provided by AgRISTARS Conservation and Pollution Project 691-08-01.			
16. Abstract  This report contains all user and programmer information required for using the Correlation Area Method (CAM) program.  CAM is a method for combining measurements of hydrologic variables from all measurement technologies to produce estimated areal mean values. The method accounts for sampling geometries and measurement accuracies and provides a measure of the accuracy of the estimated mean areal value.  The concepts of CAM were developed and published in NASA Contractor Report, "Combining Remotely Sensed and Other Measurements for Hydrologic Areal Averages."  The work reported is a part of a three phased study for NASA to develop and test technology for improving the usefulness of hydrologic modeling by incorporating remotely sensed measurements in hydrologic operational procedures.			
17. Key Words (Selected by Author(s)) Remote sensing, Hydrology, Models, Forecasting		18. Distribution Statement  Unlimited Unclassified STAR Category 48, 43	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price*